Kantronics Data Engine

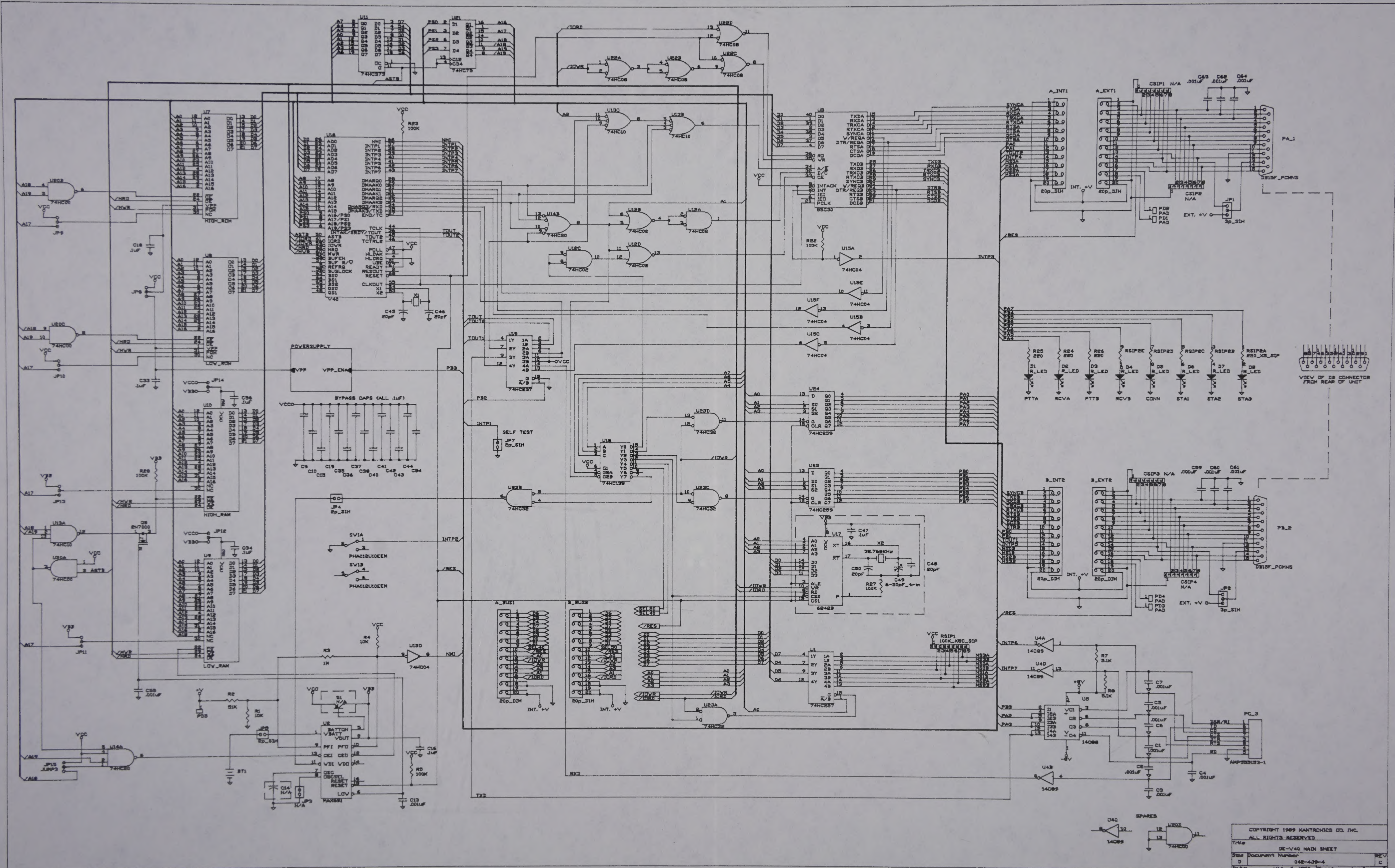
Operator's Manual

Phil Anderson
President

Jeff
Compliments
Phil

Kantronics
RF Data Communications Specialists

1202 E. 23rd Street
Lawrence, Kansas 66046
(913) 842-7745
FAX (913) 842-2021



Title		
DE-V40 MAIN SHEET		
Size: D	Document Number: 048-439-4	REV: C
Date: June 6, 1990	Sheet 1 of 2	

Kantronics Data Engine

Operator's Manual

Kantronics
RF Data Communications Specialists

1202 E. 23rd Street, Lawrence, Kansas 66046
Order number (913) 842-7745
Service number (913) 842-4476
BBS number (913) 842-4678
9 am - noon, 2 pm - 5 pm Central Time, Monday-Friday

Kantronics Data Engine

Operator's Manual

Kantronics
RF Data Communications Specialists
1202 E. 38th Street, Lawrence, Kansas 66046
Order number (913) 842-7745
Service number (913) 842-4179
BBS number (913) 842-4073
8 am - noon, 2 pm - 5 pm Central Time, Monday-Friday

First Printing April, 1990

The Data Engine is a Kantronics hardware and software design incorporating the AX.25 Version 2 Level 2 Packet protocol as adopted by the American Radio Relay League.

We have attempted to make this manual technically and typographically correct as of the date of the current printing. Production changes to the Data Engine may add errata or addendum sheets. We solicit your comments and/or suggested corrections. Please send to Kantronics Co., Inc., 1202 E 23rd Street, Lawrence, KS 66046.

Printed in the U.S.A.

Table of Contents

Warranty

To be sure you will receive notice of future updates or new product information, please take a moment to complete the warranty registration card and return it to us.

We do need your warranty registration on file.

Kantronics Co., Inc. warrants each TNC to be free from defects in material and workmanship under normal use and service for a period of one year after delivery to the ultimate user. Kantronics will repair or replace the TNC at our option, at no charge, should it become defective and should our examination disclose the TNC to be defective under warranty.

This warranty shall not apply to any unit that has been subject to misuse, neglect, accident due to wiring not of our own installation, or to use in violation of instructions furnished by Kantronics. This warranty will not be extended to units that have been repaired or altered outside our facilities.

This warranty does not cover broken or cracked cases or any accessory used in connection with the TNC. This warranty is in lieu of all other warranties expressed or implied. No representative or person is authorized to assume for Kantronics any other liability in connection with the sale of its products.

Return/Repair Procedures

Consult the limited warranty policy in this manual for the service provisions offered by Kantronics at no charge. This warranty is considered to be in force only when the customer has submitted his completed warranty registration within 10 days of purchase, and when the stipulations of the warranty have been met. Violations of warranty clauses will automatically void the warranty and service or repairs will be charged to the owner.

Service outside the warranty will be charged at the cost of parts, labor, and return shipping. Repaired units will be returned via UPS C.O.D. These C.O.D. charges can be avoided by including your VISA or MasterCard number with your unit to be repaired. Shipping and repair may then be charged.

When service or repairs appear necessary, it may be wise to call or write Kantronics to determine if the problem can be solved without returning the unit. Should you encounter difficulty in getting your TNC to "talk" to your computer, you may wish to perform some limited checks before calling or writing. Carefully check your wiring connections to the RS-232 port. Verify your terminal baud rate. It may be useful to perform a "Hard Reset". (See Hard Reset section.)

When calling, report the product name and ask for the Amateur Radio Service Department. Should you find it necessary to call for assistance, please have the following information available:

1. The unit name and serial number (the serial number is found on the rear panel.)

2. The firmware version number (the version number is displayed with the sign-on message of the TNC.)

If possible, you should have the TNC and your computer available to perform troubleshooting operations when you call.

The Service Department telephone hours are 9 am - noon and 2 pm - 5 pm Central Time 913-842-4476, Monday through Friday.

When writing, include a clear description of the problem, unit name, computer type, computer software used and if possible a DISPLAY listing from the TNC.

Returns to the factory for refund or exchange are strictly regulated. Any return for refund or exchange must be approved by the service department.

Radio Frequency Interference Statement

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital Device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiated radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that the interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced Radio/TV technician for help.

The user is cautioned that any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

RFI Suppression

In moving to the world of digital communications via computers, a new dimension of RFI may be encountered. In spite of the equipment manufacturers' diligence, each new piece of electronic equipment will react differently in each separate environment. Every amateur station will have its own unique layout, equipment variation, and antenna installations. Experience has shown that these differences are related to the total RF environment, and may be causative factors in RFI induced problems. The suggestions given here may assist in resolving RFI problems you may encounter in your "unique" station.

1. Use shielded cable for all connections between equipment.

2. Make all interconnecting cables as short as practical. A balance should be maintained between cable length and equipment proximity. At times simply moving the video monitor one foot further from an interface or other device will solve the "screen hash" problem.

3. Antenna runs should be kept away from equipment control lines and/or interconnecting cables. If it is necessary for such lines to cross each other they should do so at 90 degree angles.

4. Ground leads should be as short as possible and go to a GOOD EARTH GROUND.

5. Interconnecting cables appearing to act as radiators or antennas should be looped through a toroid. Be certain toroids, if used, are designed for the frequency in use.

Precautions

The Data Engine PC board is a four-layer PC board. Do not attempt any modifications to this board unless you are experienced in the repair of multi-layer circuit boards. If you damage the board, it cannot be repaired at the factory, and a new unit must be purchased.

The Data Engine is grounded through its connections to your transceiver. Make sure your transceiver is properly grounded and your computer has equal ground potential. Follow the grounding instructions in your transceiver manual.

Some Abbreviations

CTRL-x This is a two key combination. CTRL is the control key and x represents any alpha character. Press the CTRL key and while holding it down type the letter x (this can be capital or lower case, but will be shown as capital). Then release both. If your computer keyboard has no key labeled CTRL, consult your computer manual to determine which key performs the control key function.

$ preceding a number denotes a hex number (base 16)

<CR> = carriage return, $0D, decimal 13, CTRL-M

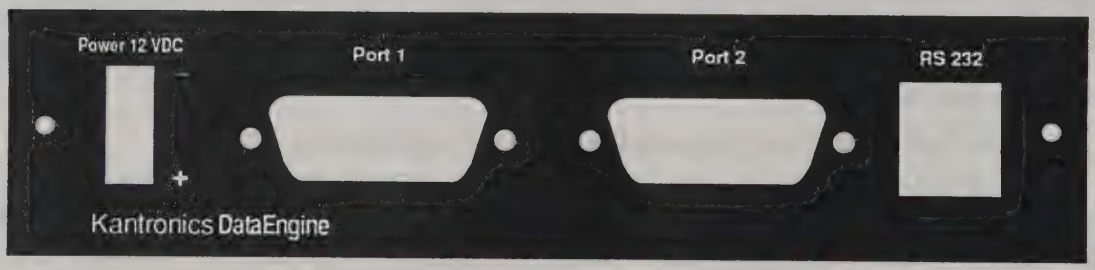
<LF> = line feed, $0A, decimal 10, CTRL-J

I/O = Input/Output

Computer and terminal are used interchangeably to describe whatever device is attached to the RS 232 port of the Data Engine.

The terms Data Engine and TNC are used interchangeably.

Back Panel



Power

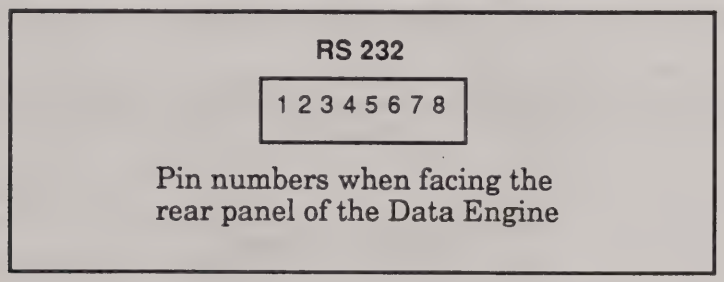
Plug the two-pin molex connector into your Data Engine port labeled Power 12 VDC, and connect the black wire from this connector to the ground (–) of a regulated 12VDC power supply. The red lead from this connector should be connected to the plus (+) side of the power supply. The Data Engine requires less than 200 ma of current.

Connecting the Data Engine to Your Computer

The Data Engine serial port uses RS-232 levels to communicate with the computer. In order to make the required connections, strip the free end of the supplied modular cable, add a connector, and attach it to your computer serial port according to the following pin-out chart.

As you look at the rear of the Data Engine, the serial port is the modular connector on the right side of the rear panel (named RS 232). The pins of the modular connector are numbered from left to right.

DE pin	Purpose	Computer DB-25 pins	Computer DB-9 pins
1	DSR	6	6
2	DCD	8	1
3	DTR	20	4
4	SG	7	5
5	RD	3	2
6	TD	2	3
7	CTS	5	8
8	RTS	4	7



Pin numbers when facing the rear panel of the Data Engine

As a minimum, you must connect pins 4 (SG), 5 (RD), and 6 (TD) from the Data Engine to your computer. This minimum cable supports only software flow control. In addition, your software program may require other lines. Refer to your software documentation for any special wiring requirements.

Normal hardware flow control requires the CTS and RTS lines to be connected (pins 7 and 8 from the Data Engine) to your computer.

Some terminal programs will also support a full RS-232 configuration. In these cases, wire all eight pins from the Data Engine to the appropriate pins on your computer serial port.

For a description of the RS-232 signals see the appendix "Definition of RS-232 signals".

Connecting the Data Engine to Your Radio

The Data Engine is connected to your radio(s) through the radio port connectors on the rear panel (Port 1 and Port 2). The exact connections required will depend on the modem you have installed in your Data Engine.

Refer to the manual that was shipped with your modem for details on the wiring of your radio to the Data Engine. If you wish to use a non-Kantronics modem, refer to the appendix "Connecting Other Modems to the Data Engine" for more details.

Assembly and Disassembly of the Data Engine

Should you require access to the Data Engine to reposition jumpers or for other purposes, disassemble as follows:

1. Turn off power to your Data Engine and remove all cables from the unit.

2. Using a small phillips screwdriver, remove the two rear-panel screws just far enough to free the panel and bezel.

3. Pull the rear-panel, bezel, and circuit board out of the case from the rear of the unit.

To reassemble, reverse the procedure above. You may wish to remove the front panel prior to inserting the circuit board into the case, as this is generally easier in lining up the LEDs.

Hard Reset

The hard reset process is provided to re-initialize the Data Engine to its factory default values. This process may become necessary should operational problems be encountered or when upgrading your firmware to a new version. This procedure is performed as follows:

1) Remove the PC board from the case as outlined in the Assembly and Disassembly section.

2) Locate jumper JP5 on the PC board. This jumper is located on the right edge of the circuit board, near the battery.

3) Remove the jumper. This disables the battery backup circuit, and the entire RAM will be erased.

4) Re-install jumper JP5 to enable the battery backup.

5) Reassamble the Data Engine and return to operation. The Data Engine will now appear as though it had just been shipped from the factory.

Now that you have your new Kantronics unit connected to your radio and computer, let's take a moment for an overview of its operations, and how you might communicate with it. The Data Engine is a Terminal Node Controller (TNC). In some ways it is very similar to a telephone modem because it receives digital signals from your computer (Terminal) and converts them to tones suitable for transmission to a distant location. The TNC also receives tones from your radio, and converts them into the digital signals understood by your computer. A TNC, however, does much more because it also controls the push-to-talk line of your transmitter, keying the radio whenever it needs to send data. It also converts data into packets, adding the required addressing, error checking, and control information to insure the data gets from one Node to the next. The error checking implemented in your TNC must be the same as the error checking used by any other station you want to talk to, and this standard method is called a protocol. The protocol used in Amateur Radio Packet TNCs is called AX.25. Different protocols are used for other modes of operation, such as AMTOR.

In order for your Data Engine to do something, you must issue instructions to it, letting it know exactly what you want done. In order to accomplish this, the Data Engine must be in Command Mode (expecting you to give it instructions) and any time you want to change the way your Data Engine operates, you must be in this mode. The Data Engine tells you that it is ready for your commands by sending you the prompt "cmd:".

When you first turn on your Data Engine out of the box (or after a hard reset), you may see some garbage characters on your screen. The Data Engine is performing an autobaud routine. It is sending the same message over and over again at different baud rates. When the Data Engine baud rate matches the baud rate set in your terminal (communications) program the display will read:

PRESS (*) TO SET BAUD RATE

When these words are readable press the asterisk, *, on your keyboard. This will set the baud rate and the Data Engine will display the following:

KANTRONICS sign-on message
ENTER YOUR CALLSIGN =>

At this point enter your amateur callsign. This callsign will be used by the Data Engine for many different things, including being in every packet you send, and deciding if a packet it receives is specifically for you. Now you should see the command prompt on your screen, as:

cmd:

The cmd: prompt means the Data Engine is ready to listen to you. Anything you type will be interpreted as a command. If the Data Engine doesn't understand, it will display:

EH?

All commands are listed alphabetically in the Commands section, and many are discussed in the following sections. Some commands require a parameter, called an argument. You would type the command, a space and then the argument (a number or whatever is appropriate).

Asynchronous Commands

Asynchronous commands are commands that allow your Data Engine and computer to talk the same language. These commands in the Data Engine will have their counterparts in your computer program, although some programs may limit what you can set. Following is a list of Data Engine command defaults and their corresponding computer settings. We will explain below, in more detail, the settings involved.

Data Engine	Computer
MODE b,p,d,s	
b = 0	Baud Rate
p = NONE	Parity no or none
d = 8	Data Bits or Word Length 8
s = 1	Stop Bits 1
ECHO ON	Full Duplex
FLOWR ON and FLOWX ON	Software Flow Control

Baud Rate is the rate that your computer and Data Engine communicate with each other. This is set in the Data Engine with the MODE command. The settings allowed by the Data Engine are 0 and continuous from 300 to 38400. The default is 0 and when left this way you will always have to go through the autobaud routine of typing the asterisk. To change this parameter, type MODE, press the space bar and type the number you wish to use. Next type the RETURN or ENTER key on your keyboard. All commands will be entered in this fashion. When you have them all set to your liking, you can PERM them (as explained later) in order to have customized defaults.

The MODE command controls many settings. When entering the command, settings are separated by commas. If you wish to leave a setting unchanged, you may omit the setting, but the comma must be present (except for trailing commas).

Word Length and Data Bits are often used interchangeably to refer to how many bits are used to recognize a character. Each character is made up of smaller elements called bits (analogous to a dit or dah in morse code). These bits are seen as high or low voltages (ones or zeroes) on the cable between the Data Engine and computer to make the desired combination for a character. A standard by the name of ASCII allows 8 bits for each character, although all the standard alphanumeric characters and punctuation can be recognized with only 7 bits. The Data Engine will talk to the computer using either 7 or 8 bits depending on the setting of the MODE command.

Parity determines what the 8th bit will be and is an old form of error detection which few modern-day programs check. Parity is none, by default, in the Data Engine which means the 8th bit will be seen as part of the character. Odd or Even Parity will change the 8th bit depending on whether there is an odd or even number of one bits in the 7-bit character. Mark and Space Parity will hold the 8th bit either high or low. In the Data Engine, parity may be set to none, even, odd, mark or space with the MODE command. (The Data Engine always sends the 8th bit in Transparent Mode).

Stop Bits are the number of bits that must be after the end of the character. The Data Engine may be set to 1 or 2.

Full Duplex or Half Duplex. (Some programs use the term Echo for Half Duplex.) When using Full Duplex the computer expects the attached device to send back (echo) what was sent to it. A setting of Half Duplex will tell the program that it must display on the screen what you type. The Data Engine's default is ECHO ON which tells the Data Engine to send back what it receives. This corresponds to a setting of Full Duplex for the computer. If you set your computer for Half Duplex you will need to turn ECHO OFF or you will see two of everything you type. When using a split-screen program you may want to set ECHO OFF in the Data Engine because the program will be handling what you type and driving the display also.

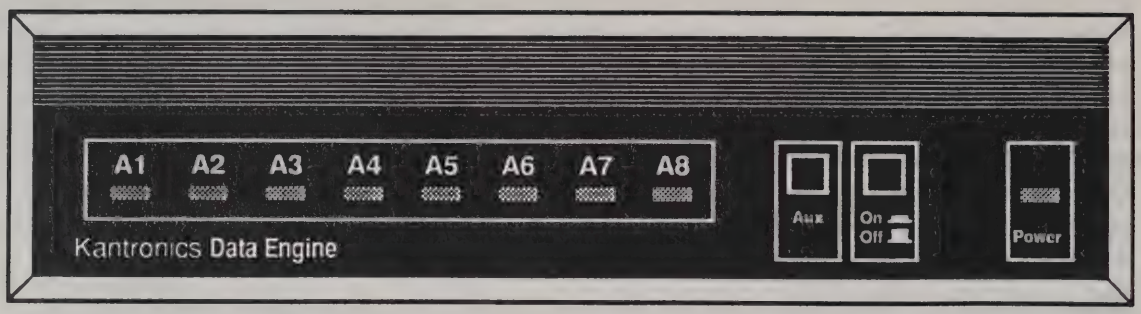
Flow Control. Often times one device may talk to another device faster than it can handle the information. When this happens the excess information is stored in a buffer until it can be processed. This buffer is only so large. If the data were to over-flow the buffer, it would be lost. Flow Control is the terminology used for how the devices inform each other to stop or start sending data. There are two ways to accomplish this, software and hardware. The Data Engine's default FLOWR ON and FLOWX ON allows the Data Engine to implement software flow control (hardware flow control is always recognized by the Data Engine).

Software Flow Control is implemented by the program and Data Engine looking at the data and watching for two certain characters. One of these characters (normally CTRL-S) tells the device to stop sending data and the other (normally CTRL-Q) tells the device to restart sending data. The Data Engine commands XOFF and XON tell the Data Engine which characters to send to the computer to stop and start data flow. The commands STOP and START define which characters the Data Engine expects from the computer. The default values for these commands are set for normal software flow control.

Hardware Flow Control is implemented by the devices watching the voltages on the RTS and CTS pins of the RS-232 port. The Data Engine will always monitor these pins so only connect them if you are going to use them. If you use hardware flow control, you should turn FLOWR OFF and FLOWX OFF. See "Connecting the Data Engine to Your Computer" for how to wire your RS-232 cable for hardware flow control.

Perm

If you would like to customize the defaults in the Data Engine, all you have to do is type PERM ALL at the cmd: prompt (followed by a RETURN or ENTER key). This command will back up your customized defaults. Then when you turn the Data Engine off and back on again, your defaults will be used. If for any reason you ever want to go back to the factory defaults a Hard Reset can be performed (see the "Hard Reset" section).



Front Panel

A1 – XMIT Port 1. This LED illuminates when the Data Engine keys the PTT line of the radio connected to Port 1.

A2 – RCV Port 1. This LED illuminates when the Data Engine detects a signal on the radio frequency of the radio connected to Port 1.

A3 – CON. This LED will illuminate when you have a packet connection on the current stream. The current stream is the stream to which the I/O is directed. (See STREAMSW in the Commands section.)

A4 – STA. This LED will illuminate when you have unacknowledged packets on the current stream.

A5 – MAIL. When the PBBS is holding mail addressed to you, this LED will be lit continuously.

A6 – unused

A7 – XMIT Port 2. This LED illuminates when the Data Engine keys the PTT line of the radio connected to Port 2.

A8 – RCV Port 2. This LED illuminates when the Data Engine detects a signal on the radio frequency of the radio connected to Port 2.

AUX – This switch is used for factory testing, and should be OUT for normal operation. Turning the unit on with this switch depressed could cause undesirable results.

ON/OFF – This switch provides power control for the Data Engine. When depressed power is applied to the Data Engine from the Power +12VDC port on the rear.

Power – This LED will illuminate when power is applied.

NOTE: If the LEDS command is OFF, only the Power LED will illuminate.

Packet Mode

Packet radio is the communication of digital data via radio. A packet is a group of characters with a flag and header at the beginning and a checksum and flag at the end. A flag is a specific character used to signify the beginning and ending of a packet. The header is information concerning who the packet is from, who it is to, any relay stations needed to get to the destination and some control information. A checksum is a complicated mathematical formula that produces a number that is unique to the combination of characters that are in the packet. This unique number is calculated by every station that handles the packet and if it does not match the number that is contained in the packet, the packet is thrown away, thus error-free communications. A packet is also called a frame.

The Terminal Node Controller (TNC) is the workhorse of packet radio. (The Data Engine is a TNC). As a listening device it hears an audio signal from the radio, changes the data to digital form, determines if it is a good packet and sends it to whatever device is attached, usually a computer. As a relay device it also checks the packets it receives and determines if the packets need to be resent, then does so if appropriate. As a sending device it receives digital data from the computer, packetizes it and changes it into audio tones which are sent out to the radio. The rules the TNC uses to do all of this is called a protocol.

The most used protocol in amateur packet radio is AX.25 Level 2 and the nitty gritty details of the inner workings can be found in a book named *AX.25 Amateur Packet-Radio Link-Layer Protocol* available from the ARRL. Many of you are not going to want to go that deep, the TNC takes care of the nitty gritty work for you, although there are parameters you can set that determine how efficiently some of that work is done. In this section of the book we will be discussing the fundamentals of how to get on the air and how parameters interrelate. The default parameters will get most everyone on the air, but by using this information you can change your parameters to be most efficient in whatever situation you find yourself.

Command Mode

In order to change parameters, or give any other instructions to the TNC you must be in Command Mode. This is the mode you will be in when you turn on the TNC (unless the INTERFACE command is set to KISS or HOST). Once you have left Command Mode for any reason there is a parameter called COMMAND that determines what special character you will use to return to Command Mode. This comes defaulted as a CTRL-C. (While holding down the control key press "c", then release both.) All parameters are described in alphabetical order in the Commands Section. Whenever you enter Command Mode the TNC will send a prompt to your screen that looks like this:

cmd:

Connected vs Unproto

There are two ways to send data in packet radio, connected or unproto (unconnected). In the Connected Mode you first establish a connection. Then your TNC will send packets to that specific station and expects acknowledgments in return. If an acknowledgment is not received, the TNC will resend the data (depending on the setting of AX25LVL it may send a poll first). The RETRY parameter will determine how many times this is done before the connection is lost due to bad conditions. If the acknowledgment is received the TNC is happy and will send more data, when available. Therefore the Connected Mode, barring impossible conditions, assures that the station you are connected to will receive everything you say, and in the order you say it.

In the Unproto Mode your TNC sends a packet. As far as the TNC is concerned the packet is not directed to a specific station therefore no acknowledgment is expected and no retrys are attempted. This mode is often used for calling CQ and informal round table chit chats.

Monitoring and Calling CQ

You will notice two callsigns at the beginning of each packet separated by a ">". The first callsign is the station the packet is from. And the second callsign is the station the packet is to. An Unproto packet may have a name or CQ for the second callsign.

To set what will be seen as the "to" callsign of an Unproto packet you use the UNPROTO command. This comes defaulted as CQ, but if you wanted to put in your name instead, you would be sure you are in Command Mode and issue a command similar to this:

u name<CR>

where u is short for unproto, name is your name and <CR> is the return or enter key on your computer keyboard. In order to call CQ you must get into the Convers Mode, so that what you are typing to the TNC will be interpreted as data to be sent out on the air and not as commands. To do this type:

k<CR>

Now anything you type will be packetized and sent out on the air. Remember to get back to Command Mode you enter a CTRL-C (default) by holding down the Control key while pressing "c". You will be going between Command and Convers Modes depending on if you want to talk to the TNC or have the TNC packetize what you type to go out on the air.

The commands MONITOR, MONLIST, MONMODE and MONTYPE determine what packets you will monitor.

A Simple Connect

Once you see a station you would like to connect to, be sure you are in Command Mode, and issue a connect request, example:

c callsign<CR>

where c is short for connect and callsign is the callsign of the station you wish to connect to. If for any reason the connection fails the TNC will send the following message to your screen:

*** RETRY COUNT EXCEEDED
*** DISCONNECTED

When your TNC does receive an acknowledgment for a connect packet it will display a message on your screen like:

*** CONNECTED TO callsign

and your TNC will change to the Convers Mode (dependent on the setting of CONMODE). Now what you type will be interpreted by the TNC as data to be sent to the other station and not commands to the TNC. The MONMODE parameter comes defaulted to NONE. Therefore once you are connected all you will see is what you type and what the other person sends you. Any packets sent by other people will not be monitored.

Two things determine when the data will be packetized. One is the parameter SENDPAC. This is defaulted as the return or enter key. As you are typing your message, whenever you hit the return or enter key you are telling the TNC to make a new packet. A second parameter, PACLEN, determines the maximum length of any packet. If you enter data longer than this length, a packet will be made, even though you have not pressed the return or enter key.

When you have finished your conversation you need to end the connection. To do this you go into the Command Mode and type a "d" for DISCONNECT. Remember to press the return or enter key after any command to the TNC. Once your station has received the acknowledgment for the disconnect packet the TNC will send this message to your screen:

*** DISCONNECTED

Either station can issue the disconnect command, no matter which station originated the connect.

Digipeating

Everything we have done so far will only be heard by those within range to hear your signal. With packet radio it is possible to get further than that. The DIGIPEAT parameter in the TNC comes defaulted ON. This makes you a possible relay station, or digital repeater – digipeater, or just digi for short. In many VHF communities one, or more, of these is put up in a good, high location and referred to as a dedicated digi. The TNC and radio is all that is needed for the digital repeater to do its job. A computer would be needed if you wanted to change a parameter, but it would not need to stay there for the digi to work. The higher the antenna, the more effective a digi will be, but remember every TNC has the capability of being a digipeater.

If we add PATH to the MONMODE command (enter MONM +PATH at the cmd: prompt) we will begin to see more than just the "from" and "to" stations of the monitored packets. We will also see the callsigns of those stations that have been used as digipeaters. (If you add HEADER to the MONMODE command (MONM +HEADER) the headers will end with a return and be on a separate line from the packet data.) This list of stations is often called a path. Here is an example of what you might see:

WK5M>KA5ZTX via IAH*,LAG,AUS:
Hi there

In this example WK5M is talking to KA5ZTX using the digipeaters IAH, LAG and AUS. The asterisk beside IAH tells you that you are hearing that digipeater. You will notice that IAH, LAG and AUS are not real callsigns. The TNC parameter MYALIAS

sets up an alias, which is often easier to remember than a callsign. To make this connection, WK5M would have typed the following command to his TNC:

c ka5ztx v iah,lag,aus

v is short for via and up to 8 digis may be used. You must specify digis in the order they will be encountered along the path from your station to the station you wish to connect to. A space must be typed after the "c" and on both sides of the "v", but digis are separated by commas and no spaces. A path can also be used with the UNPROTO command:

u cq v nom,lch,sli,bix

Unproto sets up the path for anything that is subsequently typed in the Convers Mode where no connect exists. Connect issues a connect request to the specified station, via the specified path. Then an error-free conversation can take place between the two stations.

When digipeating, the packet goes all the way from the first station, through all relay stations, then to the destination station. Then the response also has to take this same path in reverse. Chances for collisions, therefore retries, are multiplied with every digi used. This is often called end-to-end acknowledgment. Another way to get from one place to another is to connect to a "node". A node will take care of the acknowledgment between it and the next node or end user. Ask your local packeteers about other kinds of nodes which may be in your area, such as TheNet and NET/ROM, KA-NODE, Rose Switch, G8BPQ Packet Switch.

Gateway

When two modems are installed in the Data Engine a Gateway is also available. This is similar to digipeating except that the retransmission of the packet takes place on the other radio port of the TNC from where it was received.

To enable this mode, the MYGATE parameter needs a callsign that is different than any other callsign parameter.

Multi-Connects

The TNC makes it possible for you to talk to more than one person at the same time, if you want to. A stream (or channel) is used for each conversation. The command MAXUSERS determines how many streams may be used at one time. And the command USERS determines how many people can connect to you. An incoming connect uses the next available stream. If the number of streams set by USERS is full, then that station will get a busy message instead of a connect. However, if MAXUSERS is set larger than USERS, you can still issue outgoing connects on the additional streams.

The character specified in the STREAMSW parameter is used to change from one stream to another. The streams are lettered A - Z. So in order to change streams you type the STREAMSW character and then the letter designator for the stream you want (no return or enter in this case). This can be done in Command or Convers Modes.

For an example, let's assume I have my streamswitch set to a "|" (this may appear as a broken bar on your keyboard, or screen display) and I am connected to two stations: one on stream A and one on stream B. When I want to talk to the person on stream A, I type "|a" and then whatever I want to say. To talk to the person on stream B, I need to type "|b" then what I want to say. (Note that a carriage return is not required to

switch streams.) If the PORT command is set to 1 or 2, two streamswitches will be defined in the Data Engine. By default these are the "|" for Port 1 and "~" for Port 2. Using these defaults, if I wish to talk to the station connected to my A stream on Port 1 I would type "|a". To talk to the station on my C stream of Port 2, I would type "~c".

The STATUS command allows you to see who is on which stream, or the status of the stream, i.e. waiting acknowledgment, connect in progress, disconnected.

If you are connected and not monitoring other packets, the normal headers containing the "to" and "from" callsigns will not be shown. The setting of STREAMEV will then determine how often you see the stream designator. This parameter comes defaulted OFF, so the stream designators are only shown when a change in streams occurs. Turning this command ON will make the stream designators show on every packet. Turning STREAMCA ON will also add the callsign of the "from" station beside the stream designator.

Round Table Discussions

Several people talking together present a difficult situation for packet radio since the protocol calls for two stations to connect in order to make sure they receive each others' packets. If you wanted to be absolutely sure that everyone got everything you said you would have to connect to each person and retype everything to each person. That could get a bit cumbersome, so most people use the Unproto Mode and are aware that a collision may occur once in a while. You can usually tell by the conversation if something was missed, if you don't get an answer to a question it's probably not that he is ignoring you, but either the question or the answer got collided with.

With MONITOR ON, the MONLIST command can be used to set up your monitoring to see only those you want to see. If you like you may each want to connect to one person, then you know at least that one got what was said, but be sure MONMODE includes CONNECTED.

Timing

Dwait vs. Persistence and Slottime

When the TNC acts as a digipeater, packets that need to be relayed are retransmitted as soon as the frequency is clear. Because of the end-to-end acknowledgment of these kinds of packets it is best for an originating station to avoid colliding with digipeated packets. The TNC provides two ways to accomplish this delay. These two methods are the standard DWAIT method, or the newer PERSISTENCE/SLOTTIME algorithm. During a connect using no digis, this delay also gives the receiving station time to switch from transmit to receive.

Using the DWAIT method, once the TNC detects a clear frequency it will wait DWAIT (times 10 milliseconds) time before beginning to key-up the radio to transmit a packet. This is a packet originated by you not a digipeated packet.

The algorithm used with the PERSIST and SLOTTIME parameters helps avoid collisions by randomizing the wait time before transmitting. The more random the timing the less chance of two TNCs transmitting at the same time and colliding. Once the TNC detects a clear frequency it will wait SLOTTIME (times 10 milliseconds). Then it will generate a random number. If this number is smaller than the setting of PERSIST the TNC will transmit. If it is larger it will wait another SLOTTIME and then generate another random number and again decide whether to transmit or not. When using PERSIST and SLOTTIME you should set DWAIT to 0.

As an example, let's assume that PERSIST is set to 63, and SLOTTIME is set to 10. This value of SLOTTIME results in a random number being generated every 100 milliseconds. When the TNC sees that the channel is clear, it waits 100 ms, then generates a random number between 0 and 255 (inclusive). If, in our example, the number was 83, then the TNC would not start the keyup of the transmitter since 83 is greater than the 63 PERSIST value. Instead, it would wait an additional 100 ms, and if the channel is still clear, generate a new random number. This time, let's say it comes up with the number 27. Since this is less than the PERSIST value, we now start the keyup of the transmitter to send the packet.

Txdelay

TXDELAY should be adjusted to allow your radio sufficient time to switch from the receive mode to transmit and develop full power output. If the TNC sends the packet before the radio is at full power the beginning of the packet will be lost and no one will be able to decode it. It is a good idea to allow a little extra time for this parameter to allow the station you are talking to sufficient time to switch from his transmit mode back to receive. This is not usually necessary if you are connected through a digipeater, but if you are connected direct, this could make the difference between successful communications and no communications. The TNC sends flags during this period, so if someone has this set extra long you will hear a repetitive sound at the beginning of the packet.

Frack

Frame acknowledgment time. If the TNC expects an acknowledgment of a packet it has sent, it will wait FRACK seconds for the acknowledgment. If the acknowledgment is not received it will either send a poll or retransmit the packet, depending on the setting of AX25LVL. When digis are used, extra time is allowed for each transmission using the following equation:

FRACK * ((2 * n) + 1) seconds

where n is the number of digipeaters.

The FRACK timer begins when PTT is released (the packet has been sent) and is suspended when data carrier from the radio is present.

Retries AX.25 Level 2, Version 1 vs. Version 2

The way retries are accomplished depends on AX25LVL being 1 or 2. To explain this we will follow a conversation through its path. First lets assume station "A" is connected to station "B" with Version 1 protocol (AX25LVL 1). When station A sends a packet to station B, he expects to receive an acknowledge back indicating that station B has received the information. In order to verify that the proper packet (or frame) has been acknowledged, each frame has a number. This number is sent as a part of the frame so the receiving station knows where this packet belongs in the conversation. The frame numbers range from 0-7 and because of this, we are limited to a MAXFRAME of 7 (we do not want the same frame number reused in the same transmission). This is also

true for Version 2. If the first acknowledge is received, there is really no difference between the two versions, practically speaking. The difference shows up with retries, so let's assume that the packet did not get through on the first attempt.

Let's now assume that station A sends frame number 3 to station B. Station B does not receive the frame and therefore no acknowledge is received by station A. With version 1, the entire packet is retransmitted (with the same frame number), again to station B and this continues until station A receives an acknowledge from station B. This acknowledge can take two basic forms. The first time station B receives frame 3 he will send an acknowledge of the form "ready to receive frame 4" <rr4>. If this acknowledge is sent, and station A did not receive it, station A will again send frame 3. Since station B already received frame 3, he would acknowledge it with the form "I've already got frame number 3" <rej4>. This is also known as Reject Frame sent. This process would continue until the retry count is exceeded when, under version 1, the sending TNC will initiate a disconnect and dump the packet into the air UNPROTO. (The monitoring of the commands in < > depends on the settings of MONTYPE.)

Now let's look at the same conditions under version 2 (AX25LVL 2). Station B does not receive the frame and therefore no acknowledge is received by station A. This time, station A sends a POLL or question to station B saying, in effect, "did you receive my frame number 3?" <<RR3>>. Since station B did not receive the frame, he would respond with a "no I did not" <<rr3>>. This really says "I am ready to receive frame 3". At this point, station A, upon receiving the rr3 would immediately resend the entire frame. If station B had already received frame 3 once but the acknowledge never got to station A the question from station A for the retry would be the same. Station B's response however, would be different. He would respond with "ready to receive frame 4" <<rr4>>. If station A does not receive station B's reply this "POLL/REPLY" sequence would continue for the number of retries set in the sending TNC and if no response was received, the TNC at station A would then begin to issue connect requests to station B since there is still an outstanding packet of information. This is the major difference between version 1 and version 2. The connect attempts would then continue for the number of retries set in the TNC and if no response was received from station B after all of the above, station A would disconnect and dump the packet UNPROTO. The parameter RELINK can be turned OFF to avoid the reconnect attempt.

In either version 1 or version 2 another interesting feature of packet is the ability to automatically reestablish a connection. For instance, station A is connected to station B and has one frame outstanding. Station B disconnects without station A knowing it (perhaps a power failure, double disconnect or even a timeout (retry count exceeded)). The first time station B receives the outstanding packet from station A he will send a disconnected message to station A. When station A receives this, station A automatically issues a connect request to station B and the connection is reestablished to pass the outstanding traffic.

Flow Control

The flow control commands insure that the TNC gets everything that is sent to it by the computer and that the computer gets everything the TNC sends it. When the computer sends the TNC data the TNC stores this data in a buffer until it can packetize it, send it, and get acknowledgments. When the TNC sends the computer data it also stores it in a buffer until it can be processed, stored to disk, sent to printer, or whatever. This buffer area is only so big, if more data is sent than will fit in the buffer it is lost. To avoid this the two devices can tell each other to start and stop sending data. This is called Flow Control and can be accomplished in two ways, software and hardware. Which way you implement this depends on the capabilities of your computer communications program and personal preference. The cable between your computer and TNC must also be wired appropriately.

Software Flow Control

Software flow control sends special characters on the Transmit Data (TD) and Receive Data (RD) lines of the RS-232 cable. These are the same lines used for sending regular data between the TNC and computer. Software flow control normally sends a CTRL-S to stop data and a CTRL-Q to restart data. When a buffer gets close to full the device will send a CTRL-S and expect the other device to stop. When the buffer gets emptier it will send a CTRL-Q to tell the other device to send more data. How full or empty a buffer is when the special characters are sent is determined by the program. But, since the regular data lines are being used, a CTRL-S sent from the keyboard will also stop data. And likewise, if there is a CTRL-S in a file being sent, data flow will stop until a CTRL-Q is received.

FLOWR and FLOWX need to be turned ON for the TNC to use software flow control. XOFF determines the character sent by the TNC to stop the flow of data from the computer, and the XON character restarts the flow. The TNC expects the computer to send the STOP character to stop data and the START character to restart data. To use software flow control these commands would be set as follows: FLOWR ON, FLOWX ON, XOFF $13, XON $11, STOP $13, START $11.

In the Transparent Mode two more commands are provided that make it possible to send or receive these special characters and still use software flow control. FLOWTX controls flow control sent by the TNC to the computer and FLOWTR controls what the TNC expects from the computer. If both these commands are ON (and the above commands are set as stated) then software flow control will take place in both directions, to and from the TNC and computer. But if you are in Transparent Mode sending a file the computer is not going to be telling the TNC to stop and start since you are sending the file. But if there is a CTRL-S in the file, the TNC will think the computer is telling the TNC to stop and will not send any data to your computer until it receives a CTRL-Q (even if you have completed sending the file). To solve this problem you can turn FLOWTR OFF and send all characters and turn FLOWTX ON so the TNC will still tell the computer when to stop and restart. On the other hand, if receiving a file set FLOWTR ON and FLOWTX OFF.

Hardware Flow Control

Hardware flow control monitors the voltages on the Request To Send (RTS) and Clear To Send (CTS) pins of the RS-232 cable. Therefore these two wires must be in the cable between your TNC and computer. The TNC holds CTS high as long as it can receive data. Once its buffer gets full it pulls this line low. The computer program monitors this line and when it is pulled low knows to stop sending data. When the line is again pulled high by the TNC the computer program will restart sending data. On the other hand the computer holds RTS high as long as it can receive data and pulls it low to tell the TNC to stop sending data. The TNC always uses hardware flow control, so only wire the RTS and CTS pins if your computer program is also using hardware flow control.

Convers Mode vs. Transparent Mode

In the Convers Mode there are many special characters. To list a few:

Command	Default	Description
SENDPAC	CTRL-M	Causes a packet to be packetized
DELETE	CTRL-H	Backspace character
REDISPLAY	CTRL-R	Redisplays the keyboard buffer
CANLINE	CTRL-X	Cancels a line
STOP	CTRL-S	Stops output from TNC to computer
PASS	CTRL-V	Pass a special character

These characters are all very useful when having a packet conversation with someone. If you want to send a packet you hit the return. If you make a mistake you can backup with the delete or backspace key, or kill the whole line with CTRL-X. And if you really want to send one of these characters you can always proceed it with a PASS character.

Transparent Mode is made more for the sending of files, whether they be ASCII data files or program files. The special characters do not mean anything to the TNC, they are just characters to be put in a packet and sent to the radio. (XOFF, XON, STOP, START may be used depending on the settings of FLOWT, FLOWR, TXFLOW and TRFLOW, see the flow control section.) A SENDPAC character will not cause a packet to be packetized, instead this is controlled by a timer (PACTIME). This way short lines do not make short packets, therefore less overhead and more efficient use of the frequency. How congested the frequency is should be kept in mind when setting the PACLEN and MAXFRAME parameters.

Besides ignoring special characters, Transparent Mode also ignores the setting of MODE. The TNC acts as though MODE is set for parity none, and data bits 8, so be sure the computers on both ends of the connection are set the same. All monitor commands are treated as OFF in Transparent Mode. All you will see is what is being sent to you. You would probably want to set USERS to 1 so no one interferes with the transfer. The setting of ECHO is also ignored. Even if ECHO is ON, Transparent Mode will not echo to the attached terminal. Some programs allow for local echoing to the screen while uploading.

Getting Out of Transparent

Getting into the Transparent Mode is easy, you just type a "t" in Command Mode. But since Transparent Mode allows the sending of all characters you can not get out of Transparent Mode by just typing a CTRL-C (COMMAND character) as in Convers Mode. In order to get out of Transparent Mode you must follow a special sequence, or use a modem break if your program supports one. The special sequence must be followed precisely. This example assumes the COMMAND character is CTRL-C and CMDTIME is 1 second:

Wait at least 1 second since the last character was sent from the computer to the TNC
Type a CTRL-C
Within 1 second type a second CTRL-C
Within 1 second type a third CTRL-C
Wait 1 second and the cmd: prompt should appear

If the guard time of one second before and after the three CTRL-Cs is not there, the TNC assumes that they are data and sends them to the radio. Don't get impatient, one second can seem longer than you think it should.

General

Your Data Engine contains the Kantronics Personal Mailbox system which will allow you to leave messages for others which may be retrieved later. The personal mailbox is compatible with the large community bulletin board systems (RLI, MBL, etc) and will allow them to forward mail for you directly into your Data Engine. You may also place Personal or NTS type messages into your mailbox, and if the local Community BBS system allows, your Data Engine mailbox will forward these messages from your personal mailbox into the community system on request.

Configuring your PBBS

In order to enable your PBBS, you must set the MYPBBS callsign to be a unique callsign – that is, it cannot be the same as any other callsign in your Data Engine. You must also set the PBBS size, to allocate some RAM memory to the mailbox. This is accomplished with the PBBS command. The maximum setting allowable will depend on the amount of memory you have installed in your Data Engine. (See FREERAM command).

You may also want to set the inactivity timer (PBTIMER) for the PBBS, so that if someone connects to your PBBS, and suddenly stops sending data to the PBBS, then the system will automatically disconnect the user after a period of time. This will insure that a user doesn't tie up your Personal mailbox indefintely.

If you change the size of the mailbox, the Data Engine will automatically renumber any existing messages, beginning with number 1. If the new size is large enough for all existing messages, no messages will be lost.

At times, you may be away from your computer, and would like to switch a user into your mailbox automatically if he connects to your MYCALL. This can be accomplished by setting the CMSG command to PBBS. In order for this to operate, you must also have some message in the CTEXT. When this is done, a user who connects to your MYCALL will be sent your CTEXT first. Then, when the Data Engine receives the acknowledgement for the CTEXT, the user will be automatically connected to the PBBS. The Data Engine will then send the normal PBBS sign on ([DE1.02]), the PTEXT (if any) and the PBBS prompt.

Using the Data Engine PBBS

In order to use any Data Engine PBBS (even your own) first, get the cmd: prompt on your Data Engine, and then connect to the callsign of the PBBS. For instance, if MYPBBS is WK5M-3, I would simply type C WK5M-3. Since the PBBS is in my own Data Engine, no packets would be transmitted, but I would connect to the PBBS and receive the same prompt as if I had connected to someone elses PBBS.

When you connect to a Data Engine PBBS, you would first see the message from your TNC indicating that you are connected – *** CONNECTED TO WK5M-3. The PBBS will then send you its initial sign on message "[DE1.02]". If you have defined a PTEXT, the Data Engine will send it as the next line, and then it sends the PBBS command prompt. Example:

*** CONNECTED TO WK5M-3
[DE1.02]
PTEXT would be here (if any)
ENTER COMMAND: B,J,K #,KM,L,LM,R #,RM,S, or Help >

Using the PBBS is therefore the same, whether you are using your own PBBS or another persons PBBS. At this point, you are ready to send a message to another user, or issue any other mailbox command.

Let's assume I want to send a message to KA5ZTX. I would now use the send command:

S KA5ZTX

and the Data Engine responds with:

SUBJECT:

I now enter a short subject line:

Just a quick question

The PBBS now prompts you to enter your message:

ENTER MESSAGE--END WITH CTRL-Z OR /EX ON A SINGLE LINE

Now you enter the text of your message. To end the message and have it saved, type a CTRL-Z (hold down the control key and press Z), or type /EX. The CTRL-Z or /EX must be on a line by itself – do not type anything else on this line. When the message has been ended properly, the PBBS responds with:

ENTER COMMAND: B,J,K #,KM,L,LM,R #,RM,S, or Help >

You may now enter more mailbox commands. The commands available in the Data Engine PBBS are:

B	Causes the Data Engine PBBS to disconnect you from the mailbox
J	Sends a list of stations heard lately by the Data Engine. (If MHEARD is set to 0 this command will not be available.)
K #	Kill message number #
KM	Kill Mine
L	List all messages in the mailbox (If connected remotely, only lists those addressed to you, from you, or addressed to ALL)
LM	List all messages addressed to you
R #	Read message number #
RM	Read Mine
S	Send a message
H	Help – displays a short help menu

Reverse forwarding messages from your mailbox

The Kantronics Data Engine mailbox will allow you to enter messages which will be forwarded by full-service BBSs (RLI, MBL, etc). These messages have a special format, and can only be entered in your OWN personal mailbox. Let's suppose I want to send a message to WA4EWV who lives in Texas. I know his home BBS is WB5BBW, so I can put this message in my PBBS with the command:

S WA4EWV @ WB5BBW

Entering an @ BBS will cause the Data Engine to reverse forward this message to a full service BBS when requested by the full service BBS. In order to improve the chances of this message reaching its destination, you should always enter the message with complete hierarchical forwarding:

S WA4EWV @ WB5BBW.#STX.TX.USA.NA

Complete information on Hierarchical forwarding can be obtained from your local BBS system operator, but basically the first field after the @ symbol is the HOME BBS of the station you are trying to send a message. The next several fields (separated by periods) are the STATE (two letter postal abbreviation), country, and continent. In this case, since Texas is so large, it is sub-divided into smaller areas. These are indicated with the # symbol (in this case #STX – South Texas).

Messages entered into your mailbox in this format will be reverse forwarded to the full service BBS when requested, and the following rules apply:

If the first item after the @ symbol begins with "NTS" the message will be sent as TRAFFIC using the ST command. All other messages will be sent as PRIVATE with the SP command when they are sent to the full-service BBS. The Data Engine does not incorporate any means to initiate a BULLETIN from its PBBS.

If you attempt to send a message to ALL @ AMSAT, for instance, the local full service BBS would receive it as a PRIVATE message and not as a bulletin. As a result, it would not be accessible by anyone other than the SYSOPs of the BBSs. Although this may seem to be an inconvenience, it is necessary to help avoid over-congesting the packet network with duplicate copies of the same bulletin.

The Data Engine acts like a "smart BBS" when forwarding to or from a full service BBS. This means that it will no longer send the SUBJECT: prompt, nor will it send the ENTER MESSAGE prompt. You will also notice that when a full-service BBS connects to your PBBS, the Data Engine does not send the usual ENTER COMMAND prompt, but only the > is sent. This is designed to reduce the amount of data on the packet network, since "smart" BBSs know what is expected of them.

This page left blank intentionally

In order to operate in the Host Mode with the Kantronics Data Engine, you must first set the INTERFACE command to HOST. After this is accomplished, it will be necessary to perform a soft reset in order to enter the Host Mode. This may be accomplished by typing RESET at the cmd: prompt. If you want the Data Engine to always operate in the Host Mode, be sure to give the command PERM INTERFACE. You will also need to set the MODE command to the appropriate baud rate for your terminal. If the MODE command is not set, the Data Engine will run its normal autobaud routine, looking for an asterisk (*) from the keyboard. When the asterisk is entered, the Data Engine will then immediately enter Host Mode. While operating Host Mode, your program must use hardware flow control (RTS/CTS). Software flow control is not possible in Host Mode.

Communications Format

Host computer to Data Engine

The communications from the host to the Data Engine must occur in blocks. The block of data is delimited with a FEND character at beginning and end ($C0). If the FEND character appears within the block, the host must replace this character with a special sequence, consisting of a FESC ($DB) followed by a TFEND ($DC). One other special sequence may be required in the event a FESC ($DB) character is required in the data field. This is accomplished by the special sequence of a FESC ($DB) followed by a TFESC ($DD). These special sequences are the same used in the KISS code, as implemented by Phil Karn, KA9Q.

The next character is the command byte and will indicate the type of command being given to the Data Engine. The permissible characters in the command byte are C, D, or Q. A 'C' indicates a command which the Data Engine will interpret as if it were in the Command Mode. If the command byte is a 'D', the Data Engine will consider the data as data to be transmitted on the specified port and stream. The letter 'Q' in the command byte will cause the Data Engine to exit the Host Mode and return to Terminal Mode.

The next byte is the port byte. This byte must be used with every block of type 'D' to signify which port, 1 or 2, is to be used for transmission of the data. Type 'C' blocks must always specify this byte as either a 1 or 2, but this is only used on those commands which are specific to a port. This would include the CONNECT and DISCONNECT commands.

The fourth byte is the stream byte. This byte determines which stream (A-Z) the Data Engine will use for the data. If the stream byte is 0 for a data packet (command byte D), the data will be sent out UNPROTO. For commands that do not involve a specific port or stream, the port and stream bytes are ignored, but must be specified. In these cases, you should address port 1 and stream A.

After these four header bytes, the structure of the block for a command is exactly the same as if you were entering the command from the Terminal Mode of the Data Engine. If entering data to be transmitted, simply place the data in the following bytes.

After the data or command, terminate the information from the host with a FEND ($C0) character.

Data Engine to Host Computer

Communications from the Data Engine to the host also occurs in blocks, which are delimited at beginning and end with FEND characters ($C0).

After the beginning FEND, the next character is the status byte. A status byte 'C' is a response to a command from the host with the command byte 'C'. A status byte of 'D' indicates that the data was received on a connected stream. 'M' in the status byte means that the data in this block is the result of the monitor commands.

A status byte of 'S' is a status message caused by a change in the link state. Such messages include the *** CONNECTED TO, *** DISCONNECTED, and FRMR sent: types of messages. A special 'S' block of data consists of two FEND characters, the characters S00 and another FEND character. This indicates that the Data Engine has performed a soft reset, and all existing connections (if any) are no longer valid. This is equivalent to the Data Engine having just been turned on. A data block with the status byte 'R' is a *** CONNECT REQUEST. A block with a status byte 'T' is the result of the TRACE command. Port and stream bytes (defined below) are valid for 'D' and 'S' blocks, but only the port byte is valid for 'T', 'M' and 'R' blocks.

The port byte follows the status byte, and will contain the port number the specific information is from. This will be a '1' if the Data Engine is in single port operation, or a '1' or '2' if in dual-port operation.

The stream byte follows the port byte. The stream byte will be 'A' - 'Z' for data on connected streams. Data being sent to the host which is not connected data, will have the stream byte set to '0'.

If the Data Engine returns a 'C' status block with no data, this indicates that the command was accepted. This will occur on connect and disconnect commands.

The KISS transparency (FESC, FEND, TFEND, TFESC) described above is always applied to all 'D' blocks. In 'C' blocks, only the 'text' commands (BTEXT, CTEXT, RTEXT, PTEXT) implement the KISS transparency operation, and no other commands are expected to have to be concerned with KISS transparency.

A 'T' block from the host (TRACE information) is raw data, and not a hex dump of the received packet.

KISS Mode

The KISS Mode allows the Data Engine to act as a modem and packet assembler/disassembler (PAD). The heart of the work to be done concerning what happens to data must reside in your computer in order to use this mode of operation. The KISS code as designed by Phil Karn is implemented to support higher level protocols for sharing computer resources in a network fashion.

The most popular program using the KISS Mode of operation is TCP/IP or Transport Control Protocol/Internet Protocol. This program will allow simultaneous file transfers using FTP (File Transfer Protocol), user conversations using TELNET, and a Simple Mail Transfer Protocol (SMTP). In addition, multi-connect capability is built into the package, with the data being displayed only for the current "session". You can relate a session to an I/O stream in the normal TNC operating mode.

In the KISS Mode, the TNC simply passes all received data to your computer, and the computer program is responsible for all processing of that data, including decisions concerning routing, digipeating, and other control decisions. The TNC converts the synchronous data being received from the radio link into asynchronous data to be passed to the computer over the serial port, and converts the asynchronous data from your computer into the synchronous format suitable for radio transmission. The TNC retains the responsibility for these functions, as well as determining proper timing for channel access.

In the KISS Mode, channel access is determined by two settings in your TNC – namely PERSIST and SLOTTIME. The algorithm used to determine whether or not to transmit using this method has been shown to be considerably more sophisticated than the DWAIT method used by most standard AX.25 packet stations. The result of using the persistence algorithm is increased thruput under most channel conditions. For our explanation of this algorithm, let's assume a PERSIST setting of 63 and a SLOTTIME setting of 10. This slottime setting corresponds to 100 milliseconds.

When the TNC detects that the channel is clear and available (no carrier is detected), it starts a timer (SLOTTIME). When the timer expires (100 ms in our case) the TNC generates a random number between 0 and 255. If the generated number is equal to or less than the PERSIST value, the TNC keys up the transmitter and sends the data packet. With our setting of 63 the odds of this occurring after the first slottime are about 1 in 4. (Actually the odds are PERSIST plus 1 divided by 256.) If the TNC generated random number is greater than PERSIST, the TNC restarts the timer and waits for the timer to expire again before generating a new random number. This is repeated until the TNC gains channel access and sends its packet of information.

By carefully examining what happens, we can see that making SLOTTIME smaller will cause the TNC to generate the random number more frequently, whereas raising the PERSIST value will give a better chance (improve the odds) of transmitting the data. Through careful choice of these values, it is possible to improve data thruput while at the same time permitting shared channel usage by other packet users.

Data received from the radio is converted into asynchronous format by the TNC and sent to your computer. The data actually sent over the serial port is formatted with special control information, allowing the TNC to determine the type of data being received from the TNC.

Let's look at data from the TNC to the computer. First, all information flowing in this direction is data. No special messages are sent from the TNC to the computer in KISS Mode. The only data flowing in this direction is that received through the radio link. Every "frame" of data sent from the TNC will begin and end with a special FEND character. This character is the ASCII code $C0 (hex) or 192 decimal. The second byte of the data will be the data type, and will always be a $00. This means that the following information is data. If the data actually contains the FEND character ($C0) it will be necessary to tell the computer that the $C0 it receives is not the end of the frame, but simply is more data. This is accomplished by replacing the $C0 character with a special sequence consisting of a FESC ($DB) followed by a TFEND character ($DC). One final special sequence which could be sent from the TNC to the computer is a FESC ($DB) followed by TFESC ($DD). This is translated into $DB by the computer program.

Now, looking at data flowing in the other direction, that is from the computer to the TNC. There are five possible commands that you may need to issue to the TNC from the computer, and they basically concern setup parameters. These are commands needed to set TXDELAY, PERSISTENCE, SLOTTIME, FULLDUP, and finally, a command to exit the KISS Mode of operation. The only other data which the computer may send to the TNC in KISS Mode is data which is to be transmitted over the radio (HDLC) channel. The data coming from the computer must also begin and end with the same FEND character as is used for data coming from the TNC. All special character sequences must also be used to send the FEND, and FESC characters as data.

Each of the commands is assigned a command type number as follows:

TYPE	FUNCTION
0	Data to be transmitted
1	TXDELAY – second byte contains txdelay in 10 ms increments
2	PERSISTENCE – second byte contains persistence value
3	SLOTTIME – second byte contains slot interval
5	FULLDUP – if second byte is 0 sets fulldup mode, otherwise turns fulldup OFF
255	KISS – causes exit from KISS Mode

For example, if I want to set the TXDELAY in my KISS Mode TNC to 100 milliseconds, the computer would send the following bytes to the TNC:

C0 01 0A C0

and to send a data packet saying hello would be:

C0 00 68 65 6C 6C 6F C0

It is important to note that this data packet does not contain any addressing information, and therefore cannot be sent via AX.25 protocol. All of the addressing and formatting of the addresses must be done in the computer and sent as a data packet to the TNC.

One final sequence of value (particularly for PC compatible users) is the "Leave KISS Mode" sequence:

C0 FF C0

If for some reason, you have INTERFACE KISS and PERMED, when you turn the unit off and then on again you will be in KISS Mode. The only way to leave this would be to perform a hard reset, or use the TCP/IP command to leave KISS Mode, or to send the C0 FF C0 sequence from your keyboard. The PC compatibles offer this last opportunity by following this sequence:

Press and HOLD the ALT key. Type the numbers 192 from the numeric KEYPAD – Not the keyboard. Release the ALT key.

Press and HOLD the ALT key. Type the numbers 255 from the numeric KEYPAD – Not the keyboard. Release the ALT key.

Press and HOLD the ALT key. Type the numbers 192 from the numeric KEYPAD – Not the keyboard. Release the ALT key.

Now if the terminal program you are using sent all those characters, you will be out of the KISS Mode. Remember to change the INTERFACE command if you do not want your TNC to be in KISS Mode when you turn the unit off and then back on.

The Data Engine is capable of supporting dual-port KISS mode operation. This is accomplished by using the high nibble (4 bits) of the command byte to indicate which port to use. A non-zero high nibble, with the low nibble being the previously defined TYPE, will address Port 2, while a zero high nibble will address Port 1.

This page left blank intentionally

Commands Structure

There are many commands which affect operation of the Kantronics Data Engine. Some commands affect performance under specific conditions, some change parameters affecting general operation and others direct a one-time action.

The user changes parameters and issues instructions to the TNC by typing commands composed of English-like word abbreviations and variables which are numbers or strings of characters chosen by the user. You will probably never change some of these parameters.

Default values are stored in EPROM, and are the settings used at power-on. If you change any setting or value and PERM it, the new setting or value will be stored, and will be the value used at future power-on. Parameters which you change but do not PERM will revert to their stored values at next power on. A hard reset can be done to restore factory defaults, as described in "Hard Reset".

● Entry

A command is entered to the Data Engine by typing the command name and its argument (setting or value) in the Command Mode. The prompt for Command Mode is:

cmd:

The command and argument must be separated by a space, and the Data Engine takes action when a carriage return <CR> is typed. All command entries may be abbreviated to the shortest unique string. In the command list which follows, those required entries are denoted by capital letters.

You can examine the value of any parameter by typing the command name followed by a <CR>. A special command, DISPLAY, allows you to see the values of all parameters or groups of related parameters.

Once you go into Packet Convers Mode a CTRL-C (see COMMAND) needs to be entered to return you to the Command Mode. In the Packet Transparent Mode a special sequence is needed (see CMDTIME).

● Format

All commands are listed alphabetically. On the first line of a command will be the command name followed by any arguments required. Any optional arguments will be shown in square brackets [ ]. If the command accepts several different values, or a range of values, the permissible arguments will be shown in parenthesis ( ). The permissible arguments may also be shown separated by a vertical bar |. Timing parameters have their increments shown in square brackets. The second line will show the default value, and if it is a dual-port command. Example:

● **COMmand** arguments (permissible arguments)
default DUAL-PORT

Parameter Types

● **n** (range)

Any number within the range is permissible. The unit of measure (seconds, ms, baud, count, etc.) for the number will be described in the description. These are decimal numbers.

● **n** ($00 - $FF)

Several parameters are numerical codes for characters which perform special functions. The code is simply the ASCII character code for the desired character. (See the ASCII Chart at end of this manual.) Most of these characters have control characters as default values. Control characters are entered by holding down a special control key on the keyboard while typing the indicated key. For example, to type a CTRL-X, hold down the control key while typing an x, then release both keys. These special characters cannot be sent in a packet unless preceded by the pass character (see PASS) or unless you are operating in the Transparent Mode.

These numbers are displayed in hexadecimal (hex) form (base 16), and as the control combination (CTRL-x). They can be entered either in decimal or in hex. A hex number is distinguished from a decimal number by preceding it with a "$" prefix. The "digits" of a hex number represent powers of 16, analogous to the powers of 10 represented by a decimal number. The numbers 10 through 15 are denoted by the hex digits A through F. For example:

$1B = (1*16) + 11 = 27

Permissible values are shown as: (n = $00 - $FF). This is true if the MODE command is set for 8 data bits, as defaulted. If MODE is changed to 7 data bits, then permissible values are $00 - $7F. See the ASCII Chart at the end of this book for character codes and hex/decimal conversion.

If a streamswitch (STREAMSW) character or any other special character is defined as "$" then you will need to enter values in decimal, or precede the $ with the PASS character in order to enter hex numbers.

● **flags** ChoiceA | ChoiceB

Many parameters are "flags", meaning they have two possible values, ON and OFF, or YES and NO. All of the command descriptions show ON and OFF as the options; however YES (y) and NO (n) may be typed instead. A few parameters are really flags, but rather than indicating that something is "on" or "off", they select one of two ways of doing things. Some of these parameters have the values EVERY or AFTER indicating operating modes for data transmission. The possible choices are separated by a vertical bar. Some of the flag parameters will allow many choices, such as ON | OFF | DISC | PBBS.

● **callsigns** call-n

Several commands require callsigns as parameters. While these parameters are normally Amateur callsigns, they may actually be any collection of numbers and/or letters up to six characters; they are used to identify stations sending and receiving packets. A callsign may additionally include an "extension" (SSID, substation id), which is a decimal number from 0 to 15 used to distinguish two or more stations

on the air with the same Amateur call (such as a base station and a repeater). The callsign and extension are entered and displayed as call-ext, e.g. KØPFX-3. If the extension is not entered, it is set to -0, and extensions of -0 are not displayed by the TNC.

● **text**

There are some commands which have a parameter text string. This string can be any combination of letters, numbers, punctuations, or spaces up to 256 characters. In order to be used, all string parameters must contain at least one non-space character. You can even put characters with special meanings, such as carriage return, into the string by preceding them with the PASS character. The string ends when you type a (non-passed) carriage return. Since the string parameters are dual-port parameters, a / normally would not be placed in the string. If you need to have the / character in a string, simply type that character twice. For example, to enter a CTEXT of "Welcome to our station – Karl/Gloria" you would type the command:

ctext Welcome to our station – Karl//Gloria<CR>

● **dual port**

Entering Dual-Port Commands

Many commands in the Data Engine are "dual-port" commands. This means that you enter one parameter for each of the two ports. An example of such a command would be BEACON. You may wish to send a beacon every 30 minutes on Port 1, and a beacon every 60 minutes on Port 2.

Enter any dual port command by first typing the command name, a space, the value for Port 1, a slash, and then the value for Port 2. Enter a carriage return to enter the command.

As an example, let's say I want to send a beacon on port 1 every 30 minutes and a beacon on port 2 every 45 minutes. To enter this setting, I would type the command:

BEACON EVERY 30/EVERY 45<CR>

Another example would be to enter my callsign as WK5M for port 1 and WK5M-4 for port 2:

MYCALL WK5M/WK5M-4<CR>

Changing a port parameter

When you wish to set a dual-port type parameter, but only want to change one of the two ports, you must specify the slash to indicate which port you wish to change.

For example, if my txdelay is set for 30 on both of the two ports, and I wish to change only the txdelay for port 1, I would use the command:

TXDELAY 5/<CR>

In order to change only port 2, the proper command would have been:

TXDELAY /5<CR>

To change both ports to the same value enter:

TXDELAY 5<CR>

This page left blank intentionally

● **Autolf** ON|OFF

default ON

When ON, this command will cause a line feed ($10) to be sent to the terminal after every carriage return sent. If your terminal is double spacing each line, turn this command OFF, if it is overwriting, turn this command ON. This command will not affect data being sent to the radio port, only the display on your screen.

● **AX25lvl** n (n = 1|2)

default 2 DUAL-PORT

This command provides compatibility with all known packet units implementing AX.25 protocol. When set to 2, Level 2 Version 2 protocol is implemented and the Data Engine will automatically adapt to whichever version the connecting station is using. When set to 1, Level 2 Version 1 is implemented. Set this command to 1 if you need to digipeat through other units which do not digipeat version 2 packets. You may also find benefit from setting this command to 1 when using several digipeaters (not nodes) to send packets, or when conditions are marginal between the two stations involved. (NOTE: Changing this setting after connecting to another station will have no effect on the current connection.)

The major difference in V1 and V2 protocol is the method used to handle retries. In the connected mode, if a packet is sent and not acknowledged, Version 1 will resend the entire packet and then disconnect if the RETRY count is reached. Version 2 will first send a poll, the response to this poll will determine if the packet was received. It is possible that the ack was collided with and therefore the packet does not need to be resent. If the packet was not received it will be re-transmitted. Each time a poll is answered the RETRY count is reset to 0. If the RETRY count is reached, Version 2 will attempt to re-connect unless RELINK is OFF. If the re-connect attempt is unsuccessful, then Version 2 will issue a disconnect.

See also: RELINK, RETRY

For more information the book *AX.25 Amateur Packet-Radio Link-Layer Protocol Version 2.0 October 1984*, can be obtained from the ARRL.

● **AXDelay** n (n = 0 - 255) [10 millisecond increments]

default 0 DUAL-PORT

This command will set a period of time, in addition to TXDELAY, to wait before sending data after keying the transmitter. This delay is required when operating through a standard, full duplex repeater, in order to allow the repeater sufficient time to turn on its transmitter. This command can also be used to allow additional delay when using an external amplifier on your radio.

See also: AXHANG

● **AXHang** n (n = 0 - 255) [10 millisecond increments]
default 0 DUAL-PORT

This command will set the expected hang time of a full duplex repeater. If data has been heard within this time interval, then the AXDELAY setting will be ignored, since the repeater transmitter is still transmitting.

See also: AXDELAY

● **Beacon** [{EVERY | AFTER}] n (n = 0 - 255) [1 minute increments]
default AFTER 0 DUAL-PORT

A value of 0 disables the beacon. Setting a value greater than 0 activates the beacon under the conditions specified. If the optional keyword EVERY is used, a beacon packet will be sent every n*1 minute. If AFTER is used, a beacon packet will be sent ONCE after the specified interval with no channel activity.

The beacon frame consists of the text specified by BTEXT, in a packet addressed and routed according to the setting of BPATH.

See also: BTEXT, BPATH

● **BPath** dest [via call1, call2, ..., call8]
default BEACON DUAL-PORT

This commands sets the destination call (dest) and the path used to transmit beacons.

See also: BEACON, BTEXT

● **BReak** ON | OFF
default ON

When ON, sending a modem break signal from your computer to the Data Engine will cause the Engine to return to the command mode. This will return to command mode from either CONVERSE or TRANSPARENT.

See also: COMMAND

● **Btext** text (maximum 256 characters, including the command)
default (blank) DUAL-PORT

This command is used to specify the text to be transmitted as a beacon. Entering a single % will clear BTEXT.

See also: BEACON, BPATH

● **Canline** n (n = $00 - $FF)
default $18 (CTRL-X)

This command is used to change the cancel-line input editing character. When in Convers or Command Mode, entering a CTRL-X will cancel all characters input from the keyboard back to the last un-PASSed carriage return (unless PACTIME has expired and CPACTIME is turned on).

See also: CANPAC, CPACTIME, PASS

● **CANPac** n (n = $00 - $FF)
default $19 (CTRL-Y)

This command is used to change the cancel-packet command character. When in the Convers Mode, entering a CTRL-Y will cancel all keyboard input back to the last unpassed SENDPAC character (unless CPACTIME is turned ON and PACTIME has expired).

This character also functions as a cancel-output character when in the Command Mode. Typing the cancel-output character a second time re-enables normal output. For example, if you've told the Data Engine to do a DISPLAY, a CTRL-Y will cancel the remainder of the display, and a second CTRL-Y will re-enable the cmd: prompt after the next <CR>.

See also: CANLINE, CPACTIME, SENDPAC

● **CHeck** n (n = 0 - 255) [10 second increments]
default 0 DUAL-PORT

If n is greater than 0, a periodic check (poll) will be made to determine that a connected state still exists when no activity has occurred for n*10 seconds. This prevents a "hang-up" in a connected mode when a link failure occurs as a result of conditions beyond control of the connected stations. If n equals 0, then this timeout function is disabled. If AX25LVL is set to 1, a check timeout will initiate a disconnect.

See also: AX25LVL

● **CMdtime** n (n = 0 - 15) [1 second increments]
default 1

This command sets the time allowed for entry of required characters to escape the Transparent Mode. In order to allow escape to Command Mode from Transparent Mode, while permitting any character to be sent as data, a guard time of CMDTIME seconds is set up. After a delay of CMDTIME since the last characters were sent to the Data Engine, three COMMAND characters must be entered within CMDTIME of each other. After a final delay of CMDTIME the Data Engine will exit Transparent Mode and enter Command Mode. At this time, you should see the cmd: prompt. Example (if CMDTIME is 1 second and COMMAND is CTRL-C): wait one second, type a CTRL-C, within one second type a second CTRL-C, within one second type a third CTRL-C, WAIT one second, cmd: prompt should appear.

If CMDTIME is set to 0, the only exit from Transparent Mode is a modem break signal (if BREAK is ON).

See also: BREAK, COMMAND, TRANS

● **CMSg** ON | OFF | DISC | PBBS

default OFF DUAL-PORT

When OFF, the custom connect text stored in CTEXT will not be sent to the connecting station upon receiving a connect request. When ON, the custom string will be sent. When CMSG is set to DISC, the custom text will be sent to the connecting station, and then the Data Engine will disconnect from that station. If set to PBBS, the custom text will be sent to the connecting station, and then the connection will automatically be transferred to your PBBS. This will occur if the PBBS is available. If the PBBS is not available, the Data Engine will disconnect from the station. CTEXT must contain text in order for the DISC and PBBS functions to operate.

See also: CTEXT, PBBS, MYPBBS

● **COMmand** n (n = $00 - $FF)

default $03 (CTRL-C)

This command is used to change the Command Mode entry character. When COMMAND is set to the default value, typing a CTRL-C causes the Data Engine to return to Command Mode from Packet Convers Mode. (See CMDTIME for returning to Command Mode from Transparent Mode.)

See also: CMDTIME

● **CONMode** Conversational | Transparent | NONE

default Conversational

This command controls the mode the Data Engine will be in AUTOMATICALLY after a connect is received. The connect may result either from a connect request received, or a connect request originated by you. If the Data Engine is already in Convers or Transparent Mode when the connection is completed, the mode will not be changed. If you have typed part of a command line when the connection is completed, the mode change will not take place until you complete the command or cancel the line input. When set to NONE, no mode change will occur upon connection.

See also: CANLINE, CONNECT, CONVERS, TRANS

● **Connect** dest [via call1, call2, ..., call8]

immediate

This command will cause the Data Engine to issue a connect request to the destination call (dest), using any digipeaters specified in call1 through call8. This command can also be used to re-establish your connection on the current stream, but using a different path. Upon receipt of an acknowledgement from the distant station, the Data Engine will enter the mode specified in the CONMODE command.

If CONNECT is entered with no parameters, the status of the current stream is displayed.

See also: CONMODE

● **CONOk** ON|OFF

default ON DUAL-PORT

When ON, connect requests from other TNCs will be automatically acknowledged and a <ua> packet will be sent. The standard connect message will be output to the terminal, and the Data Engine will enter the CONMODE specified.

When OFF, connect requests from other TNCs will not be acknowledged, and a <dm> packet will be sent to the requesting station. The message "connect request: (call)" will be output to your terminal.

When operating with multiple connects allowed, the connection will take place on the next available stream. Connect requests in excess of the number allowed by the USERS command will receive a <dm> response, and the "connect request: (call)" message will be output to your terminal.

See also: INTERFACE, CONMODE, CONNECT, MAXUSERS, USERS

● **CONVers**

immediate

CONVERS has no options. It is an immediate command and will cause the Data Engine to enter the Conversational Mode from Command Mode on the current I/O stream. Any link connections are not affected.

See also: K, COMMAND

● **CPactime** ON|OFF

default OFF

When OFF and in the Convers Mode, packets are sent when the SENDPAC character is entered or when PACLEN is achieved. When ON and in the Convers Mode, packets are sent at periodic intervals determined by PACTIME. Characters are sent periodically as in Transparent Mode, but the local editing and echoing features of Convers Mode are enabled. CR should normally be OFF in this configuration, otherwise the SENDPAC character is appended at random intervals as the input is packetized by the timer.

See also: CONVERS, CR, PACLEN, PACTIME, SENDPAC, TRANS

● **CR** ON|OFF

default ON

When ON the SENDPAC character (normally carriage return) is appended to all packets sent in Convers Mode. Setting CR ON and SENDPAC $0D results in a natural conversation mode. Each line is sent when a <CR> is entered and arrives at its destination with the <CR> appended to the end of the line. To avoid overprinting, AUTOLF may need to be ON at the receiving end.

See also: AUTOLF, SENDPAC

● **CStamp** ON | OFF
default OFF

When ON, the daytime stamp is printed with all "*** CONNECTED TO" and "*** DISCONNECTED" messages on the terminal.

See also: CONNECT, DAYTIME, DAYSTRING, DISCONNECT, MSTAMP

● **CText** text (maximum 256 characters, including the command)
default (blank) DUAL-PORT

Enter any combination of characters and spaces up to a maximum length of 256. Entering a single "%" will clear CTEXT. This entry specifies the text of the automatic message to be sent in response to an accepted connect request provided that the parameter CMSG is not OFF.

See also: CMSG

● **DAYString** dayform
default dd/mmm/yy hh:mm:ss

This command will set the format for display of the date and time from the Data Engine. The format is free-form, with any text being permitted up to a total of 31 characters. The lower case characters m, d, y and s have special meaing to this command, and will be replaced with data from the software clock/calendar. The lower case character m will be replaced with the minutes the first time it appears after a lower case h. If the month is specified as a single m, months less than 10 will be displayed with a single digit. Likewise, if the day is specified as a single d, then days less than 10 will be single digit display. Entering two characters for month (mm) will force a two digit display for months less than 10, and two characters for day (dd) would force a two digit display. If the month is entered as three characters (mmm) the Data Engine will display the first three characters of the month name (FEB).

Use caution when entering real text into the display, as ALL lower case m, h, d, or s characters WILL be translated!

Some samples of possible strings and the resulting display would be:

mm/dd/yy hh:mm:ss	02/26/90 11:30:00
d.m.y h:mm:ss	26.2.90 11:30:00
d.mm.yy h:mm	26.02.90 11:30
mmm d 19yy h:ss CST	Feb 26 1990 11:00 CST
TIME: hh:mm DATE: mmm dd, 19yy	TIME: 11:30 DATE: Feb 26, 1990

See also: DAYTIME

● **DAytime** yymmddhhmm[ss]

This command will read and optionally set the software clock/calendar which displays the date and time in conjunction with the CSTAMP, MHEARD, and MONMODE TIME commands. When entering the daytime digits to set the clock, enter in pure number sequence with no spaces, dashes or slashes. For example: 900226113000 would indicate 1990, February 26, at 11:30:00 hours. If DAYTIME is entered with no parameter, the daytime is displayed in a form depending on the DAYSTRING setting.

See also: CSTAMP, DAYSTRING, MHEARD, MONMODE TIME

● **DBldisc** ON | OFF
default OFF

When OFF, only one DISCONNECT command need to be given to terminate an unsuccessful connect attempt. If you are currently connected, the normal disconnect sequence will occur. When ON, a normal disconnect sequence will always occur (you will not be disconnected until you receive an acknowledge of your disconnect or until the retry count is exceeded). A second DISCONNECT is required to force a local disconnect independent of the retry counter.

See also: DISCONNECT

● **DElete** n (n = $00 - $FF)
default $08 (CTRL-H)

This command sets the character to be used as the delete character. When this character is typed, the last input character is deleted. The most common settings are $08 (backspace) and $7F (delete).

● **DIGipeat** ON | OFF
default ON DUAL-PORT

When ON, any packet received that has MYCALL or MYALIAS in the digipeat list of its address field will be re-transmitted. Each station included in the digipeat list relays the packet in the order specified in the address field. Digipeating takes place concurrently with other TNC operations and does not interfere with normal connected operation of the station.

See also: HID, MYALIAS

● **DISCMode** NONE | COMMAND
default COMMAND

This command controls the mode that the Data Engine will enter when a Disconnect is received on the current I/O stream. When set to COMMAND, the Data Engine will return to the Command Mode if a disconnect is received on the current I/O stream. When set to NONE, the Data Engine will remain in the current state when a disconnect is received.

See also: CONMODE

● **Disconnect**
immediate

This command will initiate an immediate disconnect request on the current I/O stream. A successful disconnect results in the display of *** DISCONNECTED. If the RETRY count is exceeded while waiting for the connected station to acknowledge, the Data Engine moves to the disconnected state on that stream. Entering a second DISCONNECT command before RETRY has expired will result in an immediate disconnect on your end, however, the other station may be left thinking it is still connected to you. Disconnect messages are not displayed when the Data Engine is in the Transparent Mode. Other commands may be entered while the disconnect is in progress.

To disconnect a user from your Personal Mailbox, you would use the command DISC mypbbs (where mypbbs is the callsign you have set for the PBBS).

See also: DBLDISC, DISCMODE, RETRY, STATUS

● **DISPlay** [c]

This command causes the Data Engine to display a list of all of the parameters. You may also display only selected parameters by specifying the appropriate class identifier for that group. When using the DISPLAY command with a class, be sure to use a space between the DISPLAY command and the class. Classes of related parameters are:

(A)sync	asynchronous port parameters (TNC to computer)
(C)haracter	special Data Engine characters
(I)d	identification parameters
(L)ink	parameters affecting packet link (TNC to TNC)
(M)onitor	parameters affecting packets to be monitored
(T)iming	parameters affecting timing

Individual parameter values can be displayed by entering the command name followed by <CR>.

● **DWait** n (n = 0 - 255) [10 millisecond increments]
default 0 DUAL-PORT

This value is used to avoid collisions with digipeated packets. The Data Engine will wait n*10 milliseconds after last hearing data on the channel before it begins its own keyup sequence. The best value will be determined by experimentation, but will be a function of the keyup time (TXDELAY). This feature is made available to help alleviate the drastic reduction of throughput which occurs on a channel when digipeated packets suffer collisions. Digipeated packets are not retried by the digipeater, but must be restarted by the originating station. If all stations specify DWAIT, and the right value is chosen, the digipeater will capture the frequency every time it has data to send, since digipeated packets are sent without this delay.

Observations have proven that a better algorithm for avoiding collisions between end-user stations, while still allowing digipeaters the high-priority access they require, is achieved using persistence and slottime to determine proper transmit intervals.

See also: PERSIST, SLOTTIME

● **Echo** ON | OFF

default ON

When ON, characters received from the computer by the Data Engine are echoed back to the terminal. If you are receiving double print of characters you type on the keyboard, turn this command OFF. This corresponds to the setting in your terminal program for DUPLEX. If your program is set for full-duplex, set ECHO ON. If your program is set for half-duplex (some call it echo) then set ECHO in the Data Engine OFF. Regardless of the setting of this command, the Data Engine will not echo an X-OFF or X-ON character to the terminal when it receives a STOP or START character. Echo is disabled in Transparent Mode.

● **Flow** ON | OFF

default ON

When FLOW is ON, any character entered from the terminal will halt output to the terminal until the current packet or command is completed (by SENDPAC, PACLEN, or PACTIME). Cancelling the current input to the Data Engine or typing the REDISPLAY-line character will also cause the output to resume. FLOW will keep received data from interfering with data entry. When FLOW is OFF, received data will be "inter-leaved" with keyboard entry. If using a split-screen terminal program, you should have FLOW OFF and ECHO OFF to allow received data to be displayed while you type into the Data Engine's type-ahead buffer.

See also: CANLINE, CANPAC, CPACTIME, ECHO, PACLEN, REDISPLAY, SENDPAC

● **FLOWR** ON | OFF

default ON

When ON the Data Engine will stop sending characters to the attached terminal when it receives a STOP character, and will resume sending when a START character is received. This command is effective only in the Convers and Command modes, and must be ON to enable software flow control between the computer and Data Engine.

See also: FLOWTR, FLOWTX, FLOWX, START, STOP

● **FLOWTr** ON | OFF

default OFF

When ON the Data Engine will stop sending characters to the attached terminal when it receives a STOP character, and will resume sending when a START character is received. This command is effective only in the Transparent mode, and must be ON to enable software flow control between the computer and Data Engine.

See also: FLOWR, FLOWTX, FLOWX, START, STOP

● **FLOWTX** ON|OFF
default OFF

When ON, the Data Engine will send an XOFF character to the attached terminal when it can not accept any more data from the terminal. When the Data Engine is again able to receive characters from the terminal, it will send an XON character. This command is effective only in the Transparent mode, and must be ON to enable software flow control between the computer and Data Engine.

See also: FLOWTR, FLOWX, FLOWR, XON, XOFF

● **FLOWX** ON|OFF
default ON

When ON, the Data Engine will send an XOFF character to the attached terminal when it can not accept any more data from the terminal. When the Data Engine is again able to receive characters from the terminal, it will send an XON character. This command is effective only in the Convers and Command modes, and must be ON to enable software flow control between the computer and Data Engine.

See also: FLOWTR, FLOWTX, FLOWR, XON, XOFF

● **FRack** n (n = 1 - 15) [1 second increments]
default 4 DUAL-PORT

After transmitting a packet requiring acknowledgement, the Data Engine waits FRACK seconds before incrementing the retry counter and sending the packet again. If the retry count (specified by the RETRY command) is exceeded, the current operation is aborted. If the packet address includes relay requests (digipeaters), the time between retries is adjusted to FRACK * ((2 * m) + 1) where m is the number of intermediate relay stations specified. When the retried packet is sent, a random wait time is also added to avoid lockups where two units repeatedly collide with each other. The FRACK timer begins when PTT is released and is suspended when data carrier from the radio is present.

See also: RESPTIME, RETRY

● **FREeram**
immediate

This command will cause the Data Engine to display the number of bytes of RAM memory that are currently not assigned.

● **Help** [command]

This command, when entered without any arguments, will display a list of all of the commands available in the Data Engine. If an optional command is given, a brief desciption of the stated command is displayed.

● **HId** ON | OFF
default ON DUAL-PORT

When ON, an ID packet will be sent every 9.5 minutes, provided that packets are being digipeated through your station, or routed into your PBBS. This command should be ON if digipeating, gateway, or PBBS is enabled. If OFF, periodic identification packets will not be sent.

The ID packets will be addressed to the destination specified in the IPATH command, and will follow the path indicated in the IPATH command. The text of the ID message is specified using the ITEXT command.

See also: DIGIPEAT, IPATH, ITEXT, PBBS

● **Id**
immediate

When this command is entered, an identification packet will be forced. This command can be used to insure that your station identification is the last transmission before taking the station off the air. The ID packet is an un-numbered information <UI> frame whose data consists of your ITEXT. This packet will be addressed to the destination call specified in IPATH, and digipeated by any via addresses contained in the IPATH.

See also: IPATH, ITEXT

● **INterface** TERMINAL | BBS | HOST | KISS
default TERMINAL

This command will select the type of data communication between the Data Engine and the attached terminal. When set to TERMINAL, the Data Engine will operate with a standard telecommunications terminal program, or even a "dumb terminal". When set to BBS, the Data Engine will prevent specific messages from being sent to the attached terminal. This will allow full service BBS system, such as WØRLI, WA7MBL, and others, to operate without extraneous messages being generated by the Data Engine. When set to HOST, the Data Engine will package all data in a Host Mode format, and expects the attached terminal to package its data in the same format. (See the Host Mode Operation section for details on the Data Engine Host Mode.) When set to KISS, the KISS protocol as defined by Phil Karn (KA9Q) is implemented between the Data Engine and the attached terminal.

● **IPath** dest [via call1, call2, ..., call8]
default NONE DUAL-PORT

This command sets the destination call (dest) and the path used to transmit ID packets.

See also: ID, ITEXT

● **IText** text (maximum 256 characters, including the command)
default (blank) DUAL-PORT

This command is used to set the text that is transmitted in the data portion of every ID packet. Entering a single % will clear ITEXT.

See also: HID, ID, IPATH

● **K**
immediate

This single letter command is synonymous with CONVERS. It is included as a single-keystroke convenience for entering Convers Mode.

See also: CONVERS

● **Leds** ON
default ON
When OFF the software controlled front panel LEDS will not light, in order to conserve power.

● **LFadd** ON|OFF
default OFF

When ON, a linefeed character <LF> is added to outgoing packets following each carriage return <CR> transmitted in the packet. This function is similar to AUTOLF, except that the linefeed characters are added to outgoing packets rather than to text displayed locally. This feature will permit you to add linefeeds to outgoing packets if the station you are linked to is receiving overprinted packets on his display and has no local means to correct it. This character insertion is disabled in Transparent Mode.

See also: AUTOLF

● **LIdlist** NONE|call[,call, ..., call]|{+|-}call (1 - 10 calls)
default NONE DUAL-PORT

Any stations listed in the LIDLIST will be completely ignored by the Data Engine. Stations may be added to the list by specifying a + before the call, or deleted from the list by specifying a - before the call.

● **MAxframe** n (n = 1 - 7)
default 4 DUAL-PORT

MAXFRAME sets an upper limit on the number of unacknowledged packets which can be outstanding at any one time. The Data Engine will send MAXFRAME number of packets in a single transmission, if they are available.

See also: PACLEN

● **MAXUsers** n (n = 1 - 26)

default 10 DUAL-PORT

This command causes the Data Engine to allocate the memory required for the maximum number of simultaneous connections you wish to allow. Each connection uses a different stream. In order to direct what you want to say to a different stream, you use the STREAMSW character, followed by the stream ID. All streams may be used for outgoing packets, but USERS sets the number that may be used for incoming connections. Changing the value of MAXUSERS will cause the Data Engine to perform a "soft reset". In order to change the current value of maxusers, you must not be connected to any station on any stream.

See also: STATUS, STREAMSW, USERS

● **MHeard** n|LONG|SHORT (n = 0 - 15)

default 10

If the optional parameter n is specified, the Data Engine will maintain a log of the last n stations heard. Entering the MHEARD command without any options results in a standard display of the stations in the MHEARD list which includes the date and time the station was last heard if time stamping is enabled. Entering the MHEARD LONG display will cause the listing to include the digipeaters being used by the stations and the destination callsigns contained in the last transmission monitored for each station. If the MHEARD SHORT command is given, only the callsigns of the last n stations will be displayed. MHEARD logging will be disabled if PASSALL is on.

See also: MONMODE TIME, PASSALL

● **MOde** [b][,[p],[d],[s]]

(b = 0|300 - 38400)
(p = NONE|EVEN|ODD|MARK|SPACE)
(d = 7|8)
(s = 1|2)

default 0,N,8,1

The MODE command is used to set the serial port parameters of the Data Engine for communication with the attached terminal. The b parameter sets the terminal baud rate, p sets the parity, d is the number of data bits, and s is the number of stop bits. Entering only a baud rate (b) will change the baud rate, leaving the other parameters unchanged. When b is set to 0, the Data Engine will perform an autobaud routine when powered up. To change the parity without any other changes, you must either enter the baud rate and the parity or enter the comma before the parity. For example, to change only the number of data bits, you could enter the command:

MODE ,,7

● **MODEM** a[,[d],[b]

(a = 0|1|2)
(d = FULL|HALF)
(b = 1 - 65535)

default 0,HALF,1200 DUAL-PORT

This command is used to configure the Data Engine to operate properly with various modems.When MODEM is entered without any parameters, the type of modem will be shown, along with the user selected parameters – a, d, b. The TYPE is automatically selected when using a Kantronics modem (such as the DE1200 modem). In order to use other modems, refer to the appendix on "Connecting Other Modems to the Data Engine".

The a parameter is an auxilliary setting, and will vary with the type of modem installed. For the Kantronics 1200 baud internal modem, the values indicate the type of carrier detect used as follows:

0 = Sine wave carrier detect on modem board.
1 = 3105 carrier detect (energy).
2 = External carrier detect circuit.

The b parameter specifies the baud rate used by the modem. Valid selections will depend on the modem installed.

The d parameter sets the duplex mode of the modem; FULL|HALF.

● **Monitor** ON|OFF

default ON DUAL-PORT

As used with the Data Engine, the term "monitor" means to display information on your terminal which is not addressed to your station.

When OFF you will display only those stations connected to you, no matter how other montior commands are set. When ON, packets will be monitored based on the settings of the MONLIST, MONMODE, and MONTYPE commands. Any station listed in the LIDLIST will not be monitored under any circumstances. The addresses in the packets are displayed along with the data portion of the packet. Callsigns (to and from fields) are separated by a ">" and the callsign extension field (SSID) is displayed if it is other than 0. All monitor functions are disabled in Transparent Mode.

See also: LIDLIST, MONLIST, MONMODE, MONTYPE

● **MONList** (1 - 10 calls)

NONE | ALL | moncalls | ALL [EXCEPT] moncalls | {+ | -}call[(TO) | (FROM)]

default ALL DUAL-PORT

In this command, "moncalls" has the general form:

callsign[(TO) | (FROM)]

The MONLIST command will allow you to determine the source and destination callsigns to be monitored by the Data Engine. When set to ALL, all packets will be displayed on the local terminal, if permitted by the other monitor commands. When set to NONE, the Data Engine will not monitor any packets, regardless of the source or destination fields. If you are connected to another station, any packets received from that station addressed to you will be output to the terminal regardless of the setting of MONLIST.

Some examples of setting the MONLIST using moncalls will follow to help you understand proper entry format.

To set a specific list of stations to be monitored, you might use the command:

MONLIST WK5M,WØXI(TO),WDØEMR(TO),KØVAY(FROM)

A station may be added into the allowed monitoring list by using the command MONLIST +call. Optionally you may specify (FROM) or (TO) immediately after the callsign to monitor only those packets FROM or TO the designated callsign.

You may also remove a station from being monitored with the command MONLIST -call.

In order to only monitor a specific list of stations, you can use the MONLIST command and specify the specific stations you wish to monitor. Suppose I wish to monitor only WØXI, WDØEMR, and WK5M. I could accomplish this with the command:

MONLIST WØXI,WDØEMR,WK5M

In addition to the above, I may want to monitor all packets addressed to MAIL, regardless of the source of those packets. This can be accomplished by the command:

MONLIST +MAIL(TO)

Another example might be that I wish to monitor all packets except for the local Bulletin Boards. The calls of the Bulletin Boards I do not want to monitor are WBØAEX, WØXK, and KØVAY. The command used to do this would be:

MONLIST ALL EXCEPT WBØAEX,WØXK,KØVAY

As a final example, let's say that I want to monitor any packets from WØXI, and any packets addressed to MAIL, and any packets going to or coming from WK5M. The following command will accomplish this:

MONLIST WK5M,WØXI(FROM),MAIL(TO)

I could also have accomplished the same end result with the following three commands, entered at different times:

MONLIST WK5M

MONLIST +WØXI(FROM)

MONLIST +MAIL(TO)

See also: MONITOR

● **MONMode** NONE | ALL | mode | {+ | -}mode

default NONE DUAL-PORT

In this command, "mode" can be any of the following:

CONNECTED, FILTER, HEADER, PATH, TIME.

This command is used to determine what information will be monitored. If you are connected to another station, any packets received from that station addressed to you will be output to the terminal regardless of the setting of MONMODE.

If set to NONE, the Data Engine will only monitor packets when you are not currently connected to another station, and will display only the source callsign, destination callsign, and any information contained in the data field of the frame.

The modes available, and their function, are as follows:

CONNECTED	Monitor packets when connected to another station
FILTER	Strip all control characters from monitored packets except for <CR> and <LF> characters
HEADER	The Data Engine will output a newline sequence to the terminal after the address information and before the data portion of a frame received.
PATH	The Data Engine will display the complete path information contained in the address field. The source address would appear first, followed by the ">" character and then the destination address. If any digipeaters are contained in the address field, the Data Engine will display the word "via" followed by a comma-separated list of the digipeaters used. An asterisk (*) will indicate the station that actually transmitted the displayed packet.
TIME	The Data Engine will display a time stamp on the monitored packets.

See also DAYTIME, MONITOR, STREAMCA, STREAMEV

● **MONType** NONE | ALL | types | {+ | -}type
default DATA,UNPROTOS DUAL-PORT

Valid types for this command are:

COMMANDS, CONNECTS, DATA, PIDS, RESPONSES, UNPROTOS

This command determines the types of packets which will be monitored. The default value – DATA, UNPROTOS – will cause the Data Engine to monitor any I-frame between two connected stations, as well as any Un-numbered I-frame. I-frames are those frames which actually contain data a user has transmitted to another station. An Un-numbered I-frame results from a user sending data while that user is not connected to another station.

If COMMANDS is specified, the Data Engine will display the supervisory frames sent between two connected TNCs, which are used to check the validity of a link, or to query the other station concerning whether a previously sent packet was successfully received.

If CONNECTS is specified, the Data Engine will send CONNECTS <C>, DISCONNECTS <D>, DISCONNECTED MODE <dm>, and UN-NUMBERED ACKNOWLEDGEMENTS <ua> to the attached terminal for monitoring.

Specifying DATA will cause the Data Engine to monitor information containing frames between other connected users. (Data being sent in a connected state to your station is not considered monitored data, and will always be sent to your terminal.)

If PIDS is specified, the Data Engine will send information to your terminal which is contained in any received frame, regardless of the Protocol ID (PID) in use. When not specified, you will only monitor information which has a PID of $F0. This is pure AX.25, end user information. Some examples of other PIDS in use that you probably would not want to monitor are:

$CC	TCP/IP
$CD	TCP/IP
$CF	Net/Rom, TheNet, or G8BPQ Packet Switch

If RESPONSES is specified, the Data Engine will display the supervisory frames sent between two connected TNCs which are used to acknowledge information frames or other COMMAND frames.

If UNPROTOS is specified, the Data Engine will send Un-numbered I-frames (Unproto data <UI>) to the terminal for monitoring.

See also: MONITOR

● **MYAlias** call[-n] | NONE
default NONE DUAL-PORT

This command is used to enter an alias into the Data Engine which may be used for digipeating.

See also: MYCALL, MYGATE, MYPBBS, MYREMOTE

● **MYcall** call[-n]

DUAL-PORT

This command is used to enter your callsign into the Data Engine.

When the Data Engine is first turned on out of the box, or after a hard reset, it asks you for your callsign – there is NO DEFAULT. The callsign you enter is placed in this parameter. The extension n is called a Substation ID (SSID) and is defaulted as 0, but may be any number from 0 to 15. All packets originated by the Data Engine will contain this callsign in the FROM address field. Any packets received by the Data Engine with this callsign in the TO address field or digipeat fields will be responded to appropriately (connect, disconnect, ack, digipeat, etc).

See also: MYALIAS, MYGATE, MYPBBS, MYREMOTE

● **MYGate** call[-n]|NONE

default NONE

DUAL-PORT

This command is used to enter a callsign or alias into the Data Engine which may be used as a gateway between the two radio ports.

See also: MYCALL, MYALIAS, MYPBBS, MYREMOTE

● **MYPbbs** call[-n]|NONE

default NONE

DUAL-PORT

This command is used to enter a callsign or alias into the Data Engine which may be used to access the Personal Mailbox.

See also: MYCALL, MYALIAS, MYGATE, MYREMOTE, PBBS

● **MYRemote** call[-n]|NONE

default NONE

DUAL-PORT

This command is used to enter a callsign into the Data Engine which may be used to access the commands of the Data Engine remotely. Once entered a soft reset must be given to activate the desired change.

This will allow you to change the configuration of the Data Engine. Use CAUTION when accessing the remote, as it is possible to change parameters to the point that the Data Engine will no longer communicate. If this occurs, you will have to connect a terminal to the unit locally. Also, be aware that some commands may cause the Data Engine to perform a soft reset – for instance, changing the size of the PBBS. This would cause ALL connected streams to be disconnected. Since access to the command set of the Data Engine will allow ALL parameters to be changed, a password scheme (see RTEXT) is used to verify that the user attempting to connect to the remote is properly authorized.

See also: MYCALL, MYALIAS, MYGATE, MYPBBS, RESET, REMTIMER, RTEXT

● **Paclen** n (n = 1 - 256)
default 128 DUAL-PORT

This command specifies the maximum length of the data portion of a packet. The Data Engine will automatically send a packet when the number of input bytes reaches n. This value is used in both the Convers and Transparent Modes.

See also: MAXFRAME

● **PACTime** [EVERY|AFTER] n (n = 0 - 255) [1 second increments]
default AFTER 1

This parameter is always used in Transparent Mode, and will also be used in Convers Mode if CPACTIME is ON. When AFTER is specified, bytes are packaged when input from the terminal stops for n seconds. When EVERY is specified, input bytes are packaged and queued for transmission every n seconds. A zero length packet is never produced, and the timer is not started until a new byte is entered. If EVERY or AFTER is not given, the current state is retained.

See also: CPACTIME, TRANS

● **PASs** n (n = $00 - $FF)
default $16 (CTRL-V)

This command selects the ASCII character used for the "pass input" editing command. You may use this character to send any character in a packet while in Convers Mode, even though that character may have a special function. For example, if you wish to send a dollar sign ($) in a packet, but your STREAMSW is set to $24 ($), you can do so by preceding it with the PASS character. The character will be sent rather than being interpreted as a STREAMSW by the Data Engine.

In Transparent Mode, all charracters are passed, there are no special functions except the one combination to get out of Transparent Mode.

● **PASSAll** ON|OFF
default OFF

When OFF, packets will only be displayed if the CRC (error checking) is correct, and according to monitor commands. When this command is ON, the Data Engine will display packets, regardless of whether or not the CRC is correct. The Data Engine will attempt to decode the address field as well as the data field and display the packets as specified by other commands such as MONTYPE. The entire packet, determined by the beginning and ending flags, must be received before an attempt is made to decode. If both flags are not received, the data will not be decoded. MHEARD logging is disabled when PASSALL is ON.

See also: MONTYPE, MHEARD

● **PBbs** n (maximum depends on RAM)
default 0

Setting n greater than 0 allocates memory and activates the Personal Mailbox in the Data Engine. The amount of memory allocated will be n Kilobytes, and may be limited by the MAXUSERS command. Changing the size of the PBBS memory allocation will not affect the contents of the mailbox (messages will be preserved) so long as sufficient memory remains allocated to store the existing messages. Using the PBBS n command with n greater than 0 will ALWAYS renumber the messages in the mailbox beginning with message number 1. This command causes a soft reset if n is different from its previous value.

See also: CMSG, FREERAM, MYPBBS, MAXUSERS, PBTIMER, PBPERSONAL, PTEXT

● **PBPersonal** ON|OFF
default OFF

When OFF the personal mailbox will allow messages to be sent to any callsign. When ON only messages addressed to the MYCALL or MYPBBS callsigns will be accepted.

See also: MYCALL, MYPBBS, PBBS

● **PBTimer** n (n = 0 - 10) [1 minute increments]
default 10

The PBTIMER will set the amount of time the PBBS will remain connected to a user if no data is being received. After n minutes of no activity, the PBBS will automatically disconnect from the user. Setting PBTIMER to 0 disables this function.

See also: PBBS

● **PErm** ALL|parameter
immediate

This command causes the specified parameter or ALL parameters to be made "permanent"; the value(s) being PERMed are written into the battery backed-up RAM. When the unit is turned on, it checks the RAM and reloads any parameters it finds. Care should be taken when PERMing a parameter, as this cannot be undone by turning the unit off and then on again. For instance, if you PERM the INTERFACE command to KISS, the unit will immediately enter the KISS mode when turned on, and the only way out of the KISS mode would be the special KISS escape sequence, or a hard reset. To return to factory defaults a hard reset must be performed as described under "Hard Reset".

If you change a parameter, but do not PERM it, you may then recall the last PERMed setting by using the RESTORE command.

See also: RESTORE

● **PERSist** n (n = 0 - 255)

default 64 DUAL-PORT

n is used to determine if a packet will be sent after SLOTTIME expires. For example, let's assume a PERSIST setting of 63 and a SLOTTIME setting of 10. This slottime setting corresponds to 100 milliseconds. When the Data Engine detects that the channel is clear and available (no carrier is detected), it starts a timer (SLOTTIME). When the timer expires (100 milliseconds in our case) the Data Engine generates a random number between 0 and 255. If the generated number is equal to or less than the PERSIST value, the Data Engine keys up the transmitter and sends the data packet. With our setting of 63, the odds of this occurring after the first slottime are 1 in 4. (Actually the probability is PERSIST plus 1 divided by 256.) If the Data Engine generated random number is greater than PERSIST, the Data Engine re-starts the timer and waits for the timer to expire again before generating a new random number. This is repeated until the Data Engine gains channel access and sends its packet of information.

The algorithm used to determine whether or not to transmit using the PERSIST/SLOTTIME method has been shown to be considerably more sophisticated than the DWAIT method used by most standard AX.25 packet stations. The result of using the persistence algorithm is increased thruput under most channel conditions. Making SLOTTIME smaller will cause the Data Engine to generate random numbers more frequently, whereas raising the PERSIST value will give a better chance (improve the odds) of transmitting the data. Through careful choice of these values, it is possible to improve data thruput while at the same time permit shared channel usage by other packet stations. The persistence algorithm has been added on top of the DWAIT algorithm.

See also: SLOTTIME

● **POrt** n (n = 0|1|2)

default 0

If n is 0, the Data Engine will operate only through radio port 1. Radio port 2 will be disabled in this configuration. Setting n to 1 will enable both radio ports, and when first turned on, the current I/O stream will be on port 1. If n is set to 2, both ports are again enabled, but the active I/O port when the unit is turned on will be port 2.

● **PText** text (maximum 256 characters, including the command)

default (blank)

This entry specifies the customized text sent with the initial PBBS (Personal Mailbox) sign-on message (when a remote station connects to the PBBS). Enter any combination of characters and spaces up to a maximum length of 256 characters. Entering a single "%" will clear PTEXT. If you plan to "reverse forward" to a full-service BBS, do not place a > character anywhere in the PTEXT.

See also: PBBS

● **Redisp** n (n = $00 - $FF)

default $12 (CTRL-R)

This command is used to change the REDISPLAY-packet input editing character. The parameter n is the ASCII code for the character you want to type in order to REDISPLAY the packet currently being entered.

You can type this character to cause the Data Engine to re-display the packet you have begun. When you type the REDISPLAY-packet character, the following things happen: First, type-in flow control is released (if FLOW was enabled). This displays any incoming packets that are pending. Then a \ (backslash) character is displayed, and the packet you have begun is redisplayed on the next line. If you have deleted and retyped any characters, only the final form of the packet will be shown. You are now ready to continue typing where you left off. Incoming packets will continue to be displayed until you type the next character to be inserted into the packet.

You can use this character if you are typing a message in Convers Mode and a packet comes in. You can see the incoming message before you send your packet, without cancelling your input.

See also: CANLINE, CANPACK, FLOW

● **RELink** ON | OFF

default ON DUAL-PORT

When ON the Data Engine operating with AX25LVL 2 will attempt to automatically reconnect after RETRY is exceeded. When OFF the Data Engine operating with AX25LVL 2 does not attempt to automatically reconnect. The PBBS will never attempt to reconnect regardless of the setting of this command. If using AX.25 Level 2 Version 1 (AX25LVL 1) this command has no effect.

See also: AX25LVL, RETRY

● **REMtimer** n (n = 0 -10) [1 minute increments]

default 2

This command sets the amount of time a station may stay connected to the MYREMOTE without sending any data. After n minutes, the Data Engine will force a disconnect from the MYREMOTE if no data has been received from the connected station.

In addition, if the proper password is not entered within three (3) attempts, the MYREMOTE will disconnect the current user and will not accept any connects for 15 minutes.

See also: MYREMOTE, RTEXT

● **RESEt**

immediate

This command is used to perform a soft reset. Any parameters changed but not PERMed are retained. The contents of the mailbox are preserved, and the MHEARD log is not cleared. Any existing connections will no longer be recognized by the Data Engine, even though the other end still believes it is connected to you. The initial sign-on message will be displayed.

See also: MAXUSERS, MYREMOTE, PBBS, RTEXT

● **RESptime** n (n = 0 - 255) [100 millisecond increments]
default 5 DUAL-PORT

The number specified establishes a minimum delay, in 100 millisecond increments, that is imposed on acknowledgement of information-bearing packets (I-frames). Delay may run concurrently with DWAIT (PERSIST and SLOTTIME) and any other random delays in effect. This command is useful in avoiding collisions during such activity as file transfers using full-length packets.

See also: FRACK

● **RESTore** ALL|parameter
immediate

This command will set the specified parameter to the value last PERMed. This allows you to change any parameter to a new value, and still be able to recall the last PERMed value. If ALL is specified, the Data Engine will recall ALL parameters from the battery backed-up RAM. This is the same condition you would have if you had turned the Data Engine off and then on again.

See also: PERM

● **RETry** n (n = 0 - 15)
default 10 DUAL-PORT

This command specifies the number of packet retries. Packets are re-transmitted n times before the operation is aborted. The time between retries is specified by the command FRACK.

See also: AX25LVL, FRACK

● **RIng** ON|OFF
default ON

When ON, three bell characters ($07) are sent to the terminal with each "*** CONNECTED TO" message when a connect request is received from another station.

● **RText** text (maximum 256 characters, including the command)
default (blank)

This string is used to develop the authorization for access to the MYREMOTE. A long string should be placed in this parameter if the MYREMOTE is going to be used. When a station connects to the MYREMOTE, the Data Engine will generate a series of six (6) random numbers between 1 and the length of this string. The six numbers will be sent to the connected user, and the Data Engine will not allow remote access unless the user correctly decodes the six numbers. These numbers correspond to the position of the characters in the RTEXT string. The user will be given a maximum of three attempts to decode the string. If the user fails to decode the string properly, the Data Engine will disconnect the user and start the 15 minute penalty timer. No new connects to the MYREMOTE are possible until the timer expires.

Case is significant when entering the decoded string, and spaces must ONLY be entered if they are a part of the string and the generated number is the location of a space. Once entered, a soft reset must be given to activate the desired change.

See also: MYREMOTE, REMTIMER, RESET

● **SEndpac** n (n = $00 - $FF)
default $0D (<CR>)

This command specifies a character that will force a packet to be sent in the Convers Mode. In the Convers Mode, packets are sent when the SENDPAC character is entered or when PACLEN is achieved.

See also: CPACTIME, CR

● **SLottime** n (n = 0 - 255) [10 millisecond increments]
default 10 DUAL-PORT

n specifies the amount of time between successive tries of the persistence algorithm.

See also: PERSIST

● **STARt** n (n = $00 - $FF)
default $11 (CTRL-Q)

This command specifies the character sent by the terminal to the Data Engine to restart input from the Data Engine. If set to $00, only hardware flow control will be used. For software flow control, set this parameter to the character the computer will send to restart data flow.

See also: FLOWX, STOP, XOFF, XON

● **Status** [LONG]
immediate

This command will display both the identifier and link state of any currently connected streams. The current input and output (IO) stream is also indicated. A pound sign (#) indicates that there is unacknowledged data in the buffers for that stream. If LONG is specified, all streams are shown in the listing.

See also: MAXUSERS, PBBS, STREAMSW

● **STOp** n (n = $00 - $FF)
default $13 (CTRL-S)

This command specifies the character sent by the computer to the Data Engine to stop input from the Data Engine. If set to $00, only hardware flow control will be used. For software flow control set this parameter to the character the computer will send to stop data flow.

See also: FLOWX, START, XOFF, XON

● **STREAMCa** ON|OFF
default OFF

When monitoring packets addressed only to you, setting this command ON will enable the display of the callsign of the connected-to station following the stream identifier of the connection (controlled by STREAMEV). This is especially useful when operating with multiple connections allowed.

See also: MONMODE CONNECTED, MONITOR, STREAMEV

● **STREAMEv** ON|OFF
default OFF

When OFF, the stream indicator is displayed only when a change in streams occurs. When ON, the stream indicator will be displayed with every incoming packet. This command takes effect when monitoring only those packets addressed to you.

See also: MONMODE CONNECTED, MONITOR, STREAMCA, STREAMSW

● **STReamsw** n (n = $00 - $FF)
default $7C (|) DUAL-PORT

This command selects the character(s) to be used to signify that a new "stream" or connection channel is being addressed. The default character (|) may appear as a broken vertical line on your keyboard or screen. To change streams, you must type the streamswitch character followed immediately by the stream designator. It is not necessary to enter the return key after entering this combination. The stream designator is an alphabetic character A through Z limited by the value of MAXUSERS.

If STREAMSW is set to a dollar sign ($24) you will need to enter numerical code type parameter values in decimal, or precede the $ with the PASS character in order to enter hex numbers.

The character selected can be PASSed in the Convers Mode by using a special PASS character, and will always be passed as data in the Transparent Mode. If operating in the Transparent Mode and you wish to change streams, you must first return to the Command Mode.

See also: MAXUSERS, PASS, STATUS

● **TRACe** ON|OFF
default OFF

When ON, all received frames are displayed in their entirety, in hexadecimal, including all header information. All packets which are also eligible for monitoring will be displayed in normal text.

● **Trans**
immediate

This command causes immediate exit from Command Mode into Transparent Mode. The current link state is not affected. The 8th bit is sent out as received from the terminal, no matter what the parity is set to in the MODE command. Parity settings in the sending and receiving computers should be set the same for meaningful communications. There are no special editing characters, all characters are sent out as received. To get out of Transparent Mode, if the BREAK command is ON, send a modem break signal, or use the special sequence described under CMDTIME. If BREAK is OFF, refer to the CMDTIME command for details on exiting the Transparent Mode. This mode is effective for file transfers, but would not normally be used for keyboard to keyboard conversations.

See also: BREAK, CMDTIME, CONMODE

● **TRIes** n (n = 0 - RETRY – 1)

The TRIES command will display and optionally set the number of attempts which have been made to resend a packet which failed to reach its destination. For instance, if RETRY is set to 10, TRIES will show how many attempts have already been made to pass the data. For example, if TRIES were to show 8 "TRIES 3" would reset the counter to make the TNC believe that it had only tried 3 times so far, thus allowing 7 more attempts before the RETRY limit is exceeded.

See also: RETRY

● **TXdelay** n (n = 0 - 255) [10 millisecond increments]
default 30 DUAL-PORT

This command sets the transmitter key-up delay as 10*n milliseconds. This setting establishes the time delay between the application of push-to-talk and AFSK data tones to the transmitter. Flags (character to begin packet) are sent during the delay. This command needs to be set long enough to give your transmitter time to develop full power before data is sent. If set too short, the beginning of the packet will be chopped off and another station will never be able to decode your packets. If set too long, additional flags at the beginning (heard as a repetitive sound) just wastes air time. It may be necessary to increase your TXDELAY to allow the receiving station sufficient time for his receiver to detect your signal (i.e. switch from transmit back to receive).

See also: AXDELAY, AXHANG

● **Unproto** dest [via call1[,call2,...,call8]]

default CQ DUAL-PORT

In this command, dest is the destination address (this is really just a dummy address as no connection takes place). Some people often put their first name or CQ here.

Call1 through call8 are the optional stations which are supposed to relay the packets. Up to 8 digipeaters may be specified. This is referred to as a path.

This command is used to set the digipeat and destination address fields for packets sent in the unconnected (unprotocol) mode. Unproto packets do not receive an acknowledgement and are not retried. They are sent as Unsequenced I-frames <UI>. If dest is set to NONE, no unconnected packets will be sent except for BEACON and ID packets. Unconnected packets sent from other units can be monitored by setting MONITOR ON, and MONTYPE to UNPROTOS.

See also: BEACON, BPATH, ID, IPATH

● **UPLOAD**

immediate

This command allows you to load information from the serial port into the RAM memory. For further details, we suggest you purchase the Data Engine Developer's Manual.

● **USers** n (n = 0 - 26)

default 1 DUAL-PORT

This command specifies the channels (streams) which may be available to incoming connect requests. For example, if USERS = 5, then an incoming connect request will connect to the lowest channel A-E, if any of these channels are in the unconnected state. If none of the 5 channels are available (all of them are connected), a <dm> packet will be sent back to the requesting station and the message "*** connect request: (call)" will be output to your terminal. If USERS is set to 0, no one will be able to connect to you. If USERS is set higher than MAXUSERS, the extra is ignored and the message "USERS LIMITED BY MAXUSERS" will be displayed.

See also: MAXUSERS, STREAMSW

● **Xoff** n (n = $00 - $FF)
default $13 (CTRL-S)

This command selects the character sent by the Data Engine to the terminal to stop input from the terminal. If set to $00, hardware flow control must be used. For software flow control, set this parameter to the character the computer expects to see to stop sending data to the Data Engine.

See also: FLOWX, XON

● **XON** n (n = $00 - $FF)
default $11 (CTRL-Q)

This command selects the character sent by the Data Engine to the terminal to restart input from that device. If set to $00, hardware flow control must be used. For software flow control, set this parameter to the character the terminal expects to see to restart sending data to the Data Engine.

ASCII Chart

Ctrl	Dec	Hex	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex
@	0	00	NUL	32	20	SP	64	40	@	96	60	`	128	80	160	A0	192	C0	224	E0
A	1	01	SOH	33	21	!	65	41	A	97	61	a	129	81	161	A1	193	C1	225	E1
B	2	02	STX	34	22	"	66	42	B	98	62	b	130	82	162	A2	194	C2	226	E2
C	3	03	ETX	35	23	#	67	43	C	99	63	c	131	83	163	A3	195	C3	227	E3
D	4	04	EOT	36	24	$	68	44	D	100	64	d	132	84	164	A4	196	C4	228	E4
E	5	05	ENQ	37	25	%	69	45	E	101	65	e	133	85	165	A5	197	C5	229	E5
F	6	06	ACK	38	26	&	70	46	F	102	66	f	134	86	166	A6	198	C6	230	E6
G	7	07	BEL	39	27	'	71	47	G	103	67	g	135	87	167	A7	199	C7	230	E7
H	8	08	BS	40	28	(	72	48	H	104	68	h	136	88	168	A8	200	C8	232	E8
I	9	09	HT	41	29	)	73	49	I	105	69	i	137	89	169	A9	201	C9	233	E9
J	10	0A	LF	42	2A	*	74	4A	J	106	6A	j	138	8A	170	AA	202	CA	234	EA
K	11	0B	VT	43	2B	+	75	4B	K	107	6B	k	139	8B	171	AB	203	CB	235	EB
L	12	0C	FF	44	2C	,	76	4C	L	108	6C	l	140	8C	172	AC	204	CC	236	EC
M	13	0D	CR	45	2D	-	77	4D	M	109	6D	m	141	8D	173	AD	205	CD	237	ED
N	14	0E	SO	46	2E	.	78	4E	N	110	6E	n	142	8E	174	AE	206	CE	238	EE
O	15	0F	SI	47	2F	/	79	4F	O	111	6F	o	143	8F	175	AF	207	CF	239	EF
P	16	10	DLE	48	30	0	80	50	P	112	70	p	144	90	176	B0	208	D0	240	F0
Q	17	11	DC1	49	31	1	81	51	Q	113	71	q	145	91	177	B1	209	D1	241	F1
R	18	12	DC2	50	32	2	82	52	R	114	72	r	146	92	178	B2	210	D2	242	F2
S	19	13	DC3	51	33	3	83	53	S	115	73	s	147	93	179	B3	211	D3	243	F3
T	20	14	DC4	52	34	4	84	54	T	116	74	t	148	94	180	B4	212	D4	244	F4
U	21	15	NAK	53	35	5	85	55	U	117	75	u	149	95	181	B5	213	D5	245	F5
V	22	16	SYN	54	36	6	86	56	V	118	76	v	150	96	182	B6	214	D6	246	F6
W	23	17	ETB	55	37	7	87	57	W	119	77	w	151	97	183	B7	215	D7	247	F7
X	24	18	CAN	56	38	8	88	58	X	120	78	x	152	98	184	B8	216	D8	248	F8
Y	25	19	EM	57	39	9	89	59	Y	121	79	y	153	99	185	B9	217	D9	249	F9
Z	26	1A	SUB	58	3A	:	90	5A	Z	123	7A	z	154	9A	186	BA	218	DA	250	FA
[	27	1B	ESC	59	3B	;	91	5B	[	124	7B	{	155	9B	187	BB	219	DB	251	FB
/	28	1C	FS	60	3C	<	92	5C	\	124	7C	|	156	9C	188	BC	220	DC	252	FC
]	29	1D	GS	61	3D	=	93	5D	]	125	7D	}	157	9D	189	BD	221	DD	253	FD
^	30	1E	RS	62	3E	>	94	5E	^	126	7E	~	158	9E	190	BE	222	DE	254	FE
_	31	1F	US	63	3F	?	95	5F	_	127	7F	DEL	159	9F	191	BF	223	DF	255	FF

This page left blank intentionally

In Case of Difficulty

Power light fails to light

Check to be sure the Data Engine is connected to a 12 volt DC supply and that the correct polarity has been observed (red wire to positive, black wire to negative).

Check to be sure the power supply is turned on and operating properly. Measure the output voltage from the power supply. It should be between 11 and 14 volts DC for the Data Engine to operate properly.

Check to be sure the front panel on-off switch is depressed.

LEDs (A1-A8) fail to operate

Check to be sure the LEDS command is set to ON. The software controlled LEDs will not operate with this command turned OFF.

Will not communicate with computer

Check the wiring of the supplied modular connector to your computer with your manual for the computer serial port and this manual.

Be sure the cable is firmly seated at both ends.

Test your computer serial port by placing a jumper between the RD and TD lines of the computer serial port. When you run your terminal program and type on the keyboard with this jumper in place, everything you type should be echoed back to your receive data area of the terminal program.

Before you call the factory

We suggest that you contact other packet users in your area, and also computer experts to assist in determining that your computer is properly operating.

Check to be sure the AUX switch is in the out position.

This page left blank intentionally

TXD Transmit Data. This line is the serial data from the terminal which is to be transmitted to the other station by the TNC. It is this line which is used for all communication from your terminal to the TNC, including commands.

RXD Receive Data. This line is used by the TNC to send the data it receives from the other station to your terminal. This line is also used to send TNC messages to your terminal.

SG Signal Ground. This line establishes the common reference potential for all circuits except Protective Ground.

RTS Request To Send. This line tells the TNC that the terminal is ready to receive data. An ON level tells the TNC it may send data while an OFF level tells it to stop sending data. If the terminal for any reason is unable to accept data from the TNC, it will cause this line to change to an OFF state, providing that the terminal supports hardware flow control. For instance, buffer is full, terminal is turned off, and so on.

CTS Clear To Send. This line is used by the TNC to tell the terminal whether or not it may send data to the TNC. An ON level tells the terminal it may send data while an OFF level tells it to stop sending data. This pin is the complement to the RTS pin, implementing hardware flow control in the other direction.

DCD Data Carrier Detect. This line is an output from the TNC indicating connected status of the TNC. When a connection exists on the current stream, this line will be ON.

DSR Data Set Ready. Some terminal programs check this pin to see that the TNC is operating before allowing you to talk to the TNC. This pin is pulled ON when the Data Engine is turned on.

DTR Data Terminal Ready. Most terminals will supply an ON voltage to this pin when powered up and capable of receiving data. This pin is not checked by the Data Engine.

Cable Wiring

Transmit Data (TD), Receive Data (RD) and Signal Ground (SG) must always be wired in order for the TNC and the computer to exchange any data. Many terminal programs also require the use of hardware flow control from the TNC. For hardware flow control, Request To Send (RTS) and Clear To Send (CTS) must also be wired. Check the documentation to your terminal program to see if any other wires are required.

Some programs want to see Data Set Ready (DSR) to know that the TNC is there before operating. If this is the case wire both DSR and Data Terminal Ready (DTR). Sometimes you can satisfy the program's needs by jumpering these two pins at the computer end of the cable. Data Carrier Detect (DCD) is needed by some BBS software to know that a connection has taken place. This would require wiring DCD. Some phone modem programs also want to see a connection before allowing you to even talk to the TNC. This case can usually be solved by jumpering DCD to DTR at the computer end of the cable. If your computer requires DSR and also DCD, it is perfectly acceptable to jumper all three pins (DTR, DSR, and DCD) together on the computer end of the cable.

The TNC is wired as DCE (Data Communication Equipment). DCE equipment always sends its data on the RD wire. DTE (Data Terminal Equipment) talks on TD. This means that if a computer is wired internally as DCE and attached to the TNC it will need to have TD from the computer wired to RD on the TNC, and RD from the computer wired to TD of the TNC. Otherwise they will both be talking on the same wire and never hear what is said. If properly implemented by the DCE computer, hardware flow control may be used by connecting RTS from each device to CTS on the other device.

DB-25 Connector

Male (Looking at Pins)

Female (Looking at Holes)

DB-9 Connector

Male (Looking at Pins)

Female (Looking at Holes)

Connecting Other Modems to the Data Engine

The Data Engine will accept modems from other manufacturers, including the Texnet, K9NG, G3RUH, HAPN, and TAPR modems. Since these modems will not fit inside the Data Engine case, it will be necessary to connect them to the Data Engine through the DB-15 connector(s) on the rear panel.

In order to provide the proper signals to the DB-15 for your external modem, you will need to jumper various pins from the internal modem disconnect header to the external disconnect header (both inside the Data Engine). The pin assignments for the internal header are:

Pin	Purpose
1	Ground
2	SYNC
3	TXD
4	RXD
5	TRXC
6	RTXC
7	RTS
8	CTS
9	DCD
10	DTR
11	PB4
12	PB5
13	TOUT2
14	INTP4
15	MS0
16	MS1
17	MS2
18	MS3
19	Ground
20	+DC (Supply voltage)

The various MODEM types are selected by grounding the proper combination of MS0 - MS3. The standard Kantronics DE1200 modem is a TYPE A modem, which is selected by grounding MS0 only.

The Kantronics DE9600/G3RUH modem is automatically selected by grounding MS1, which configures the Data Engine for TYPE B modems. Type B modems supply their own transmit clock and return the receive clock to the Data Engine. As a result, the MODEM command will not display the baud rate.

The G3RUH modem and Pac-Comm PSK-1 modem require a 16X clock. This is obtained by selecting modem TYPE C (grounding MS0 and MS1). The 16X clock is then available on TOUT2. This TYPE applies to most TAPR type modems.

A type D modem is selected by grounding MS2 only. This is used for loop-back or direct wire connection to another unit.

Most external modems require the TNC to provide the Push-to-Talk signal and watchdog timer. Kantronics has a plug-in board to provide these functions and route the required signals for external modems to the DB-15 connector. The board has jumpers to allow selection of TYPE A, TYPE B, or TYPE C modems and a 15 second watchdog timer.

At present, these are the only modem TYPES supported, although other TYPES can be selected by grounding MS0-MS4 in accordance with the following chart. (Ground the pins marked X)

TYPE	MS0	MS1	MS2	MS3
A	X			
B		X		
C	X	X		
D			X	
E	X		X	
F		X	X	
G	X	X	X	
H				X
I	X			X
J		X		X
K	X	X		X
L			X	X
M	X		X	X
N		X	X	X
O	X	X	X	X

If you have, or develop a modem, which requires different signals or a new modem TYPE, please send your requirements in a fully documented letter to:

Kantronics
1202 E. 23rd St.
Lawrence, KS 66046

Data Engine Specifications

(with DE1200 Modem)

Size: 1-3/4" x 6" x 9"

Weight: 2-1/2 lbs.

Power Requirements: 12 VDC <150 ma (with DE1200 installed)

Watch Dog Timer: ≈2-1/2 minutes

External Carrier Detect (XCD): Ground active

PTT Output: Open collector PNP transistor, +40 VDC max.

Audio Output:

- Output Drive: LO = 10mv p-p, HI = 50 mv p-p, OFF = 2 v p-p
- Output Impedance: 600Ω (ac coupled)

Audio Input:

- Input Sensitivity: 20 mv
- Input Impedance (unbalanced): 600Ω
- Max Input Voltage: 2 v p-p

Other Features:

- PBBS, Gateway, Host Mode, KISS Mode

This page left blank intentionally

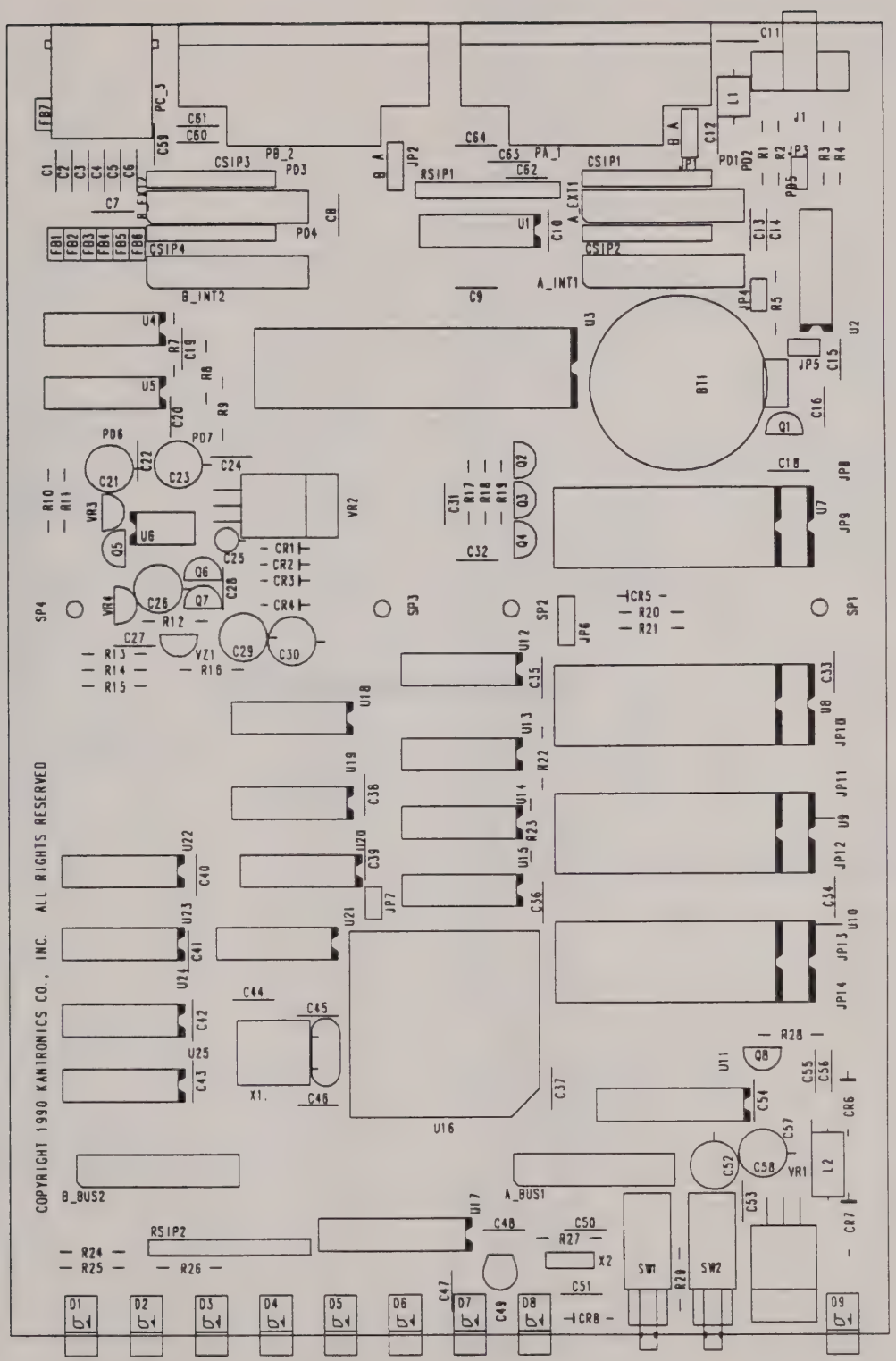
Data Engine Parts List

R1	10K
R2	51K
R3	1M
R4	10K
R5	100K
R7	100K
R8	100K
R9	47K
R10	1.8K
R11	330
R12	1K
R13	1K
R15	4.7K
R16	18K
R17	22K
R18	47K
R19	100K
R21	100K
R22	100K
R23	100K
R24	220
R25	220
R26	220
R27	100K
R28	100K
R29	220
C1	.001uF
C2	.001uF
C3	.001uF
C4	.001uF
C5	.001uF
C6	.001uF
C7	.001uF
C8	.1uF
C9	.1uF
C10	.1uF
C11	.1uF
C12	.1uF
C13	.001uF
C15	.1uF
C16	.1uF
C18	.1uF
C19	.1uF
C20	.1uF
C21	47uF_al
C23	47uF_al
C24	.1uF
C25	1uF_al
C26	47uF_al
C27	.1uF
C28	.1uF
C29	47uF_al
C30	47uF_al
C31	.1uF
C32	.1uF
C33	.1uF
C34	.1uF
C35	.1uF
C36	.1uF
C37	.1uF
C38	.1uF
C39	.1uF
C40	.1uF
C41	.1uF
C42	.1uF
C43	.1uF
C44	.1uF
C45	20pF
C46	20pF
C47	.1uF
C48	20pF
C49	6-50pF_trim
C50	20pF
C51	.001uF
C52	47uF_al
C53	.1uF
C54	.1uF
C55	.001uF
C56	.1uF
C57	.1uF
C58	47uF_al
C59	.001uF
C60	.001uF
C61	.001uF
C62	.001uF
C63	.001uF
C64	.001uF
CR1	1N914
CR2	1N914
CR3	1N914
CR4	1N914
CR5	1N914
CR6	1N4001
CR7	1N4001
CR8	1N914
D1	R_LED
D2	R_LED
D3	R_LED
D4	R_LED
D5	R_LED
D6	R_LED
D7	R_LED
D8	R_LED
D9	G_LED

L1	FER_21-200J
L2	10uH
Q1	PN2907A
Q2	2N7000
Q3	2N7000
Q4	2N7000
Q6	PN2222
Q8	2N7000
Q7	PN2907A
U1	74HC257
U2	MAX691
U3	85C30
U4	14C89
U5	14C88
U6	LTC1044
U7	27010_W/SCKT
U8	27010_W/SCKT
U9	621001LP_W/SCKT
U10	621001LP_W/SCKT
U11	74HC373
U12	74HC02
U13	74HC10
U14	74HC20
U15	74HC04
U16	V40_W/PGA_SCKT
U17	6242B
U18	74HC138
U19	74HC257
U20	74HC00
U21	74HC75
U22	74HC08
U23	74HC32
U24	74HC259
U25	74HC259
VR1	LM78M05
VR3	LM317LZ
VR4	LM317LZ
X1	19.6608MHz
X2	32.768KHz

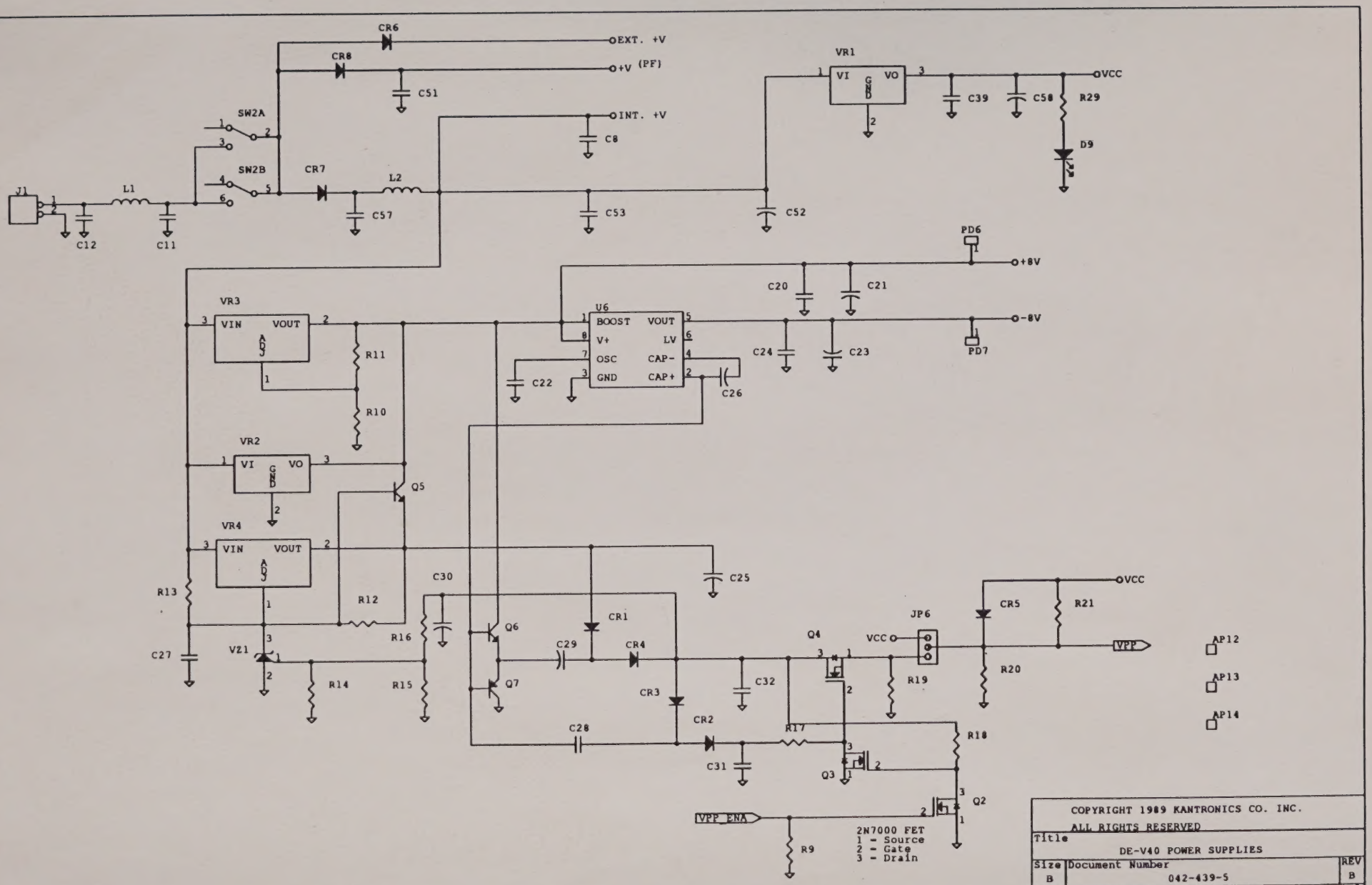
A_BUS1	20p_DIH
A_EXT1	20p_DIH
A_INT1	20p_DIH
B_BUS2	20p_DIH
B_EXT2	20p_DIH
B_INT2	20p_DIH
JP1	3p_SIH
JP2	3p_SIH
JP4	2p_SIH
JP5	2p_SIH
JP7	2p_SIH
JP6	3p_SIH
SW1	PHA012U10EEM
SW2	PHA012U10EEM
CSIP1	.001uF_X7C_SIP
CSIP2	.001uF_X7C_SIP
CSIP3	.001uF_X7C_SIP
CSIP4	.001uF_X7C_SIP
BT1	2430 Battery Clip
AP1	2430 Battery
J1	MLX39-29-1028
AP12	MLX39-00-0060
AP13	MLX39-00-0060
AP14	MLX39-01-2020
VZ1	TL431
PA_1	DB15F_PCMNS
PB_2	DB15F_PCMNS
PC_3	AMP555153-1
RSIP1	100K_X8C_SIP
RSIP2	220_X5_SIP

Data Engine Parts Location



This page left blank intentionally

Index



Title	DE-V40 POWER SUPPLIES	
Size B	Document Number 042-439-5	REV B
Date: March 16, 1990	Sheet 2 of 2